PATTERNS 2022

The Fourteenth International Conferences on Pervasive Patterns and Applications

ISBN: 978-1-61208-953-9

April 24 - 28, 2022

Barcelona, Spain

PATTERNS 2022 Editors

Joschka Kersting, Paderborn University, Germany

PATTERNS 2022

Forward

The Fourteenth International Conferences on Pervasive Patterns and Applications (PATTERNS 2022), held on April 24 - 28, 2022, continued a series of events targeting the application of advanced patterns, at-large. In addition to support for patterns and pattern processing, special categories of patterns covering ubiquity, software, security, communications, discovery and decision were considered. It is believed that patterns play an important role on cognition, automation, and service computation and orchestration areas. Antipatterns come as a normal output as needed lessons learned.

The conference had the following tracks:

- Patterns basics
- Patterns at work
- Discovery and decision patterns
- Medical and facial image patterns
- Tracking human patterns

Similar to the previous edition, this event attracted excellent contributions and active participation from all over the world. We were very pleased to receive top quality contributions.

We take here the opportunity to warmly thank all the members of the PATTERNS 2022 technical program committee, as well as the numerous reviewers. The creation of a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and effort to contribute to PATTERNS 2022. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

Also, this event could not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the PATTERNS 2022 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope PATTERNS 2022 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of pervasive patterns and applications. We also hope that Barcelona provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city

PATTERNS 2022 Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria

Adel Al-Jumaily, University of Technology, Australia
George A. Papakostas, International Hellenic University – Kavala, Greece

PATTERNS 2022 Publicity Chair

Javier Rocher, Universitat Politècnica de València, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

PATTERNS 2022

Committee

PATTERNS 2022 Steering Committee

Herwig Manaert, University of Antwerp, Belgium
Wladyslaw Homenda, Warsaw University of Technology, Poland
Patrick Siarry, Université Paris-Est Créteil, France
Yuji Iwahori, Chubu University, Japan
Alexander Mirnig, University of Salzburg, Austria
Adel Al-Jumaily, University of Technology, Australia
George A. Papakostas, International Hellenic University – Kavala, Greece

PATTERNS 2022 Publicity Chair

Javier Rocher, Universitat Politècnica de València, Spain
Lorena Parra, Universitat Politecnica de Valencia, Spain

PATTERNS 2022 Technical Program Committee

Andrea F. Abate, University of Salerno, Italy
Akshay Agarwal, IIIT Delhi, India
Carlos Alexandre Ferreira, INESC TEC, Portugal
Adel Al-Jumaily, University of Technology, Australia
Sidnei Alves De Araujo, Nove de Julho University (UNINOVE), Sao Paulo, Brazil
Zahra Ebadi Ansaroudi, University of Salerno / Foundazio Bruno Kessler, Italy
Danilo Avola, Sapienza University of Rome, Italy
Johanna Barzen, University of Stuttgart, Germany
Nadjia Benblidia, Saad Dahlab University - Blida1, Algeria
Anna Berlino, Consultant in Tourism Sciences and Valorization of Cultural and Tourism Systems, Italy
Fatma Bouhlel, University of Sfax, Tunisia
Uwe Breitenbücher, IAAS - University of Stuttgart, Germany
Alceu S. Britto, Pontifical Catholic University of Paranā (PUCPR), Brazil
Jean-Christophe Burie, L3i laboratory | La Rochelle University, France
Simone Cammarasana, CNR-IMATI, Genova, Italy
David Cárdenas-Peña, Universidad Tecnológica de Pereira, Colombia
Bidyut B. Chaudhuri, Indian Statistical Institute, India
Sneha Chaudhari, AI Organization | LinkedIn, USA
Diego Collazos, Universidad Nacional de Colombia sede Manizales, Colombia
Sergio Cruces, University of Seville, Spain
Mohamed Daoudi, Institut Mines-Telecom / Telecom Lille, France
Abhijit Das, Indian Statistical Institute, Kolkata, India
Jacqueline Daykin, King's College London, UK / Aberystwyth University, Wales & Mauritius

Moussa Diaf, Mouloud Mammeri University, Algeria
Chawki Djeddi, Université de Tébessa, Algeria
Ole Kristian Ekseth, NTNU & Eltorque, Norway
Eslam Farsimadan, University of Salerno, Italy
Eduardo B. Fernandez, Florida Atlantic University, USA
Michaela Geierhos, Research Institute CODE | Bundeswehr University Munich, Germany
Markus Goldstein, Ulm University of Applied Sciences, Germany
Eduardo Guerra, Free University of Bolzen-Bolzano, Italy
Abdenour Hacine-Gharbi, University of Bordj Bou Arreridj, Algeria
Geert Haerens, Engie, Belgium
Lukas Harzenetter, University of Stuttgart - Institute of Architecture of Application Systems (IAAS), Germany
Jean Hennebert, University of Applied Sciences HES-SO, Fribourg, Switzerland
Wladyslaw Homenda, Warsaw University of Technology, Poland
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Wei-Chiang Hong, School of Computer Science and Technology - Jiangsu Normal University, China
Kristina Host, University of Rijeka, Croatia
Marina Ivasic-Kos, University of Rijeka, Croatia
Yuji Iwahori, Chubu University, Japan
Francisco Jaime, University of Malaga, Spain
Anubhav Jain, Idiap Research Institute, Switzerland
Agnieszka Jastrzebska, Warsaw University of Technology, Poland
Maria João Ferreira, Universidade Portucalense, Portugal
Hassan A. Karimi, University of Pittsburgh, USA
Joschka Kersting, Paderborn University, Germany
Christian Kohls, TH Köln, Germany
Vasileios Komianos, Ionian University, Corfu, Greece
Sylwia Kopczynska, Poznan University of Technology, Poland
Fritz Laux, Reutlingen University, Germany
Gyu Myoung Lee, Liverpool John Moores University, UK
Reynolds León Guerra, Advanced Technologies Application Center (CENATAV), Havana, Cuba
Frank Leymann, UniversityofStuttgart, Germany
Josep Lladós, Computer Vision Center - Universitat Autònoma de Barcelona, Spain
Himadri Majumder, G. H. Raisoni College of Engineering and Management, Pune, India
Herwig Mannaert, University of Antwerp, Belgium
Pierre-Francois Marteau, IRISA / Université Bretagne Sud, France
Ana Maria Mendonça, University of Porto / INESC TEC - INESC Technology and Science, Portugal
Abdelkrim Meziane, Research Center on Scientific and Technical Information - CERIST, Algeria
Alexander Mirnig, University of Salzburg, Austria
Fernando Moreira, Universidade Portucalense, Portugal
Antonio Muñoz, University of Malaga, Spain
Dinh-Luan Nguyen, Michigan State University, USA
Hidehiro Ohki, Oita University, Japan
Krzysztof Okarma, West Pomeranian University of Technology, Szczecin, Poland
Alessandro Ortis, University of Catania, Italy
Martina Paccini, CNR-IMATI, Italy
George A. Papakostas, Eastern Macedonia and Thrace Institute of Technology, Greece
Maria Antonietta Pascali, CNR - Institute of Clinical Physiology, Italy

Copyright Information

Table of Contents

Tackling the "We have no Data`` Challenge: Domain-Specific Machine Translation in SMEs 1
Frederik S. Baumer, Sergej Denisov, Bastian Sirvend, and Jens Weber

Implicit Statements in Healthcare Reviews: A Challenge for Sentiment Analysis 5
Joschka Kersting and Frederik Simon Baumer

Concept of an Inference Procedure for Fault Detection in Production Planning 10
Jan Michael Spoor, Jens Weber, Simon Hagemann, and Frederik Simon Baumer

Track Me if You Can: Insights into Profile Interlinking on Social Networks 18
Sergej Denisov, Frederik Simon Baumer, and Michaela Geierhos

Patterns for Quantum Error Handling 22
Martin Beisel, Johanna Barzen, Frank Leymann, Felix Truger, Benjamin Weder, and Vladimir Yussupov

Contributions by Feature Layers in Two-Class Deep Learning Image Classification Decisions 31
Debanjali Banerjee and Chee-Hung Henry Chu

Tackling the ”We have no Data“ Challenge: Domain-Specific Machine Translation in SMEs

Frederik S. Bäumer & Sergej Denisov
Bielefeld University of Applied Sciences
Bielefeld, Germany
{fbaeumer1,sdenisov}@fh-bielefeld.de

Bastian Sirvend
Wonki GmbH
Bielefeld, Germany
bastian.sirvend@wonki.tech

Jens Weber
University of Applied Sciences Zwickau
Zwickau, Germany
jens.weber@fh-zwickau.de

***Abstract*—The use of translation software has decisive advantages for companies. For example, they facilitate communication and the editing and creation of multilingual documents. In contrast to the services of a translation agency, the results are immediately available and can be adapted flexibly. Nevertheless, concerns exist, especially regarding translation quality in case of specialized vocabulary, industry-specific phrases, and data security. Developing and deploying self-hosted business-specific translation models can address both problems by increasing speed and providing company-specific translations. However, this often leads to a situation where companies assume that they cannot contribute the necessary training data. In fact, many companies are sitting on a veritable treasure of data that needs to be lifted. This paper intends to show how we support enterprises with processes and software tools to create datasets for their translation solutions. For this purpose, we apply data acquisition techniques and data preparation methods, sentence alignment, and human-in-the-loop tools.**

***Index Terms*—machine learning, machine translation**

I. Introduction

Translation software is one of the tools needed daily in many companies, as it facilitates international cooperation immensely and simplifies working with multilingual documents and websites [1]. Compared to translation agencies, software has the advantage that companies can use them quickly, flexibly, and cost-effectively. However, many companies still have concerns about the quality of the translations, which is of crucial importance in corporate use since the technical vocabulary of a domain can differ considerably from that of the standard language [1]. In addition, data security concerns cause companies to be skeptical of cloud translation providers and drive the development of self-hosted translation models.

The possibility of training and operating own translation models is not a new concept. These machine translation solutions have been widely available for quite some time (e.g., [2]). The achievements of the last years, especially in the field of deep learning-based approaches, have brought the whole area of translation approaches a significant step forward [3]. In general, these machine translation systems generate translations by using statistical models that have been parameterized by analyzing documents available in the source and target languages. As a result, they can develop a basic sense of language and learn the specifics of domain-specific vocabulary. Especially low resource languages benefit from achievements such as transfer learning. However, today we face the situation that extensive resources for common languages exist, and the languages most requested can be served with existing models [1]. What remains is that these resources and models often do not reflect the peculiarities of enterprise- and domain-specific languages. However, this can be accomplished using the company's data.

While implementing in-house translation solutions, it often occurs that companies assume that they do not have their own data corpus that can be used to train the translation solutions at the very beginning. Often, companies are not aware that a parallel corpus does not necessarily have to be the starting point, but that web pages, instructions for products, advertising material, etc. can also be used to create parallel corpora – as long as they are multilingual, at least in parts. In this short paper, we report on our approach and developed software tool that makes it possible to generate necessary datasets in close cooperation with the companies.

The structure of this paper is as follows. Section II presents related work. Based on this, our approach is presented (Section III) and then discussed in Section IV. Finally, we conclude our work (Section V).

II. Related Work

For recent machine translation algorithms, underlying parallel corpora are essential, and the challenge of creating high-quality datasets is already well known. Thus, the questions of where such data comes from, who owns them, what their characteristics are and who is allowed to use them are not new either [4]. In the following, we describe existing work focusing on the mining of heterogeneous data sources to create parallel corpora for translation purposes.

Documents from parliaments or other public offices and institutions are a popular source for parallel corpora (esp. transcripts). Examples of this are *Europarl*, a corpus composed from texts from the proceedings of the European Parliament [5], and *The United Nations Parallel Corpus*, a corpus composed from United Nations documents [6]. These sources are very helpful because the documents are of very high quality, and the authors have a legitimate interest and often also the obligation to provide the identical content in several languages [6]. Furthermore, they are easy to process because they are homogeneous in format and structure [7]. This processing is different for unstructured or heterogeneous sources such

ISBN: 978-1-61208-953-9

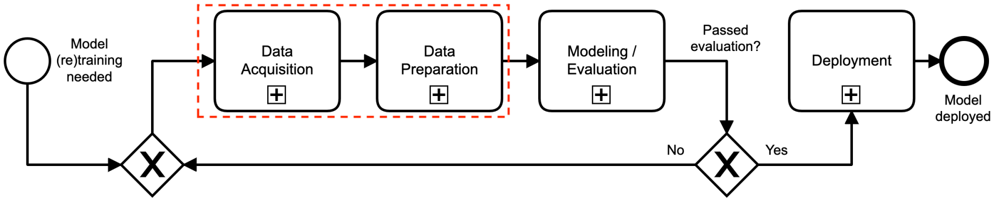


Fig. 1. Process overview for translation model development in close cooperation with the company

that more effort is required to prepare the data: An example for that is the Web. Due to the immense growth of publicly accessible multilingual websites, the Web has become an important data source for parallel corpora. A distinction must be made between approaches that consider the entire Web as a corpus [8] and approaches that use isolated pages [9] or related pages for defined subject areas [10]. Therefore, parallel corpora based on the Web exist and are widely used [11]. However, due to legal reasons not everything available online may also be used [5].

The general approach to developing parallel corpora on heterogeneous datasets is constantly evolving. It can be refined in detail depending on languages [7], sources [5], and data quality [4], but for the most part includes the following steps [5]: Obtain, Extract, Split, Prepare and Map/Alignment. A key step in this process is alignment. Alignment can be done at document, paragraph, and sentence levels. The final dataset is usually a parallel corpus at sentence level so that the sentence alignment is rarely omitted. However, alignment is not a new approach in machine translation [12]. Still, the underlying technologies have evolved immensely in recent years [13], making mass semantic alignment across hundreds of languages possible in a reasonable amount of time. Whether sentences are considered parallel depends on the alignment score. This score indicates the similarity of the source and target sentences. Existing parallel corpora often differ in the score from which sentences are considered parallel. The higher the average alignment score, the higher the quality of the training data. However, a high minimum score (e.g., 95%) leads to a significant reduction in sentences.

Less research exists in the area of business- and domain-specific parallel corpora. At the same time, commercial providers are aware of this need: The "*Microsoft Custom Translator*", for example, can be fine-tuned on datasets while using domain-specific base models (e.g., chemistry, art, agriculture). However, existing approaches can be well applied to internal company documents and domain-specific sources. A particular situation is that documents often have to be collected and merged from various platforms, data lakes, cloud services, and numerous providers and external suppliers in many different and partly proprietary formats. Ownership of the relevant documents may be subject to various departments, leading to conflicts. We will take a closer look at this situation in the following.

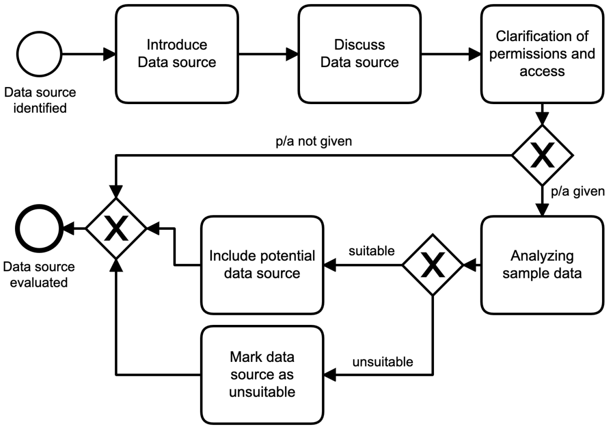


Fig. 2. Data acquisition process

III. Approach

Training an in-house translation solution requires data. The motivation of this approach is to obtain sufficient multilingual documents throughout the company, some of which are only accessible in the various functional departments. Here, a significant part is convincing people to participate, "demystifying" the process, and transparently communicating the individual steps of the necessary data acquisition and preparation.

A. Convince and involve people

Usually, individual departments place the order for a translation solution, often through internationally active units such as the "International Marketing" department. In these cases, the contact persons rarely have a technical background and often do not have permission to use the necessary data. To start the whole process, we recommend to "de-mystify" the process. This increases customer support and brings creativity to the data acquisition step and trust in the process and the results. It also makes it easier for contact persons within the company to communicate the process and obtain support.

From a process perspective, our approach (see Figure 1) at the top level is based on CRISP-DM [14]. This is a very well-known approach for data mining projects, which is already known in many companies. In the "Data Acquisition" step (see Figure 2), data sources are identified that contain potentially valuable data such as who owns this data, how it can be used, and how access can be established are examined. Once initial test data is available on the identified data source, it is

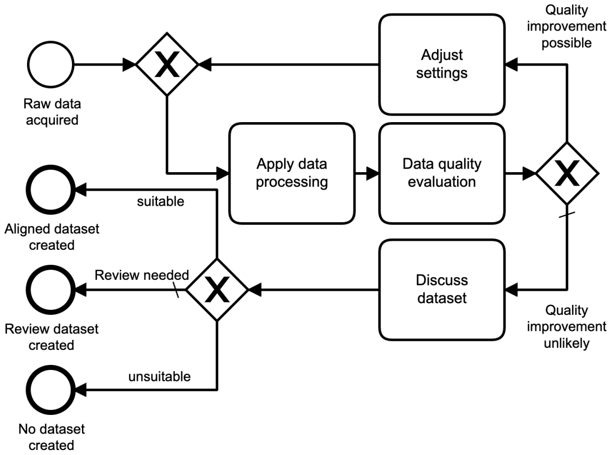


Fig. 3. Data preparation process

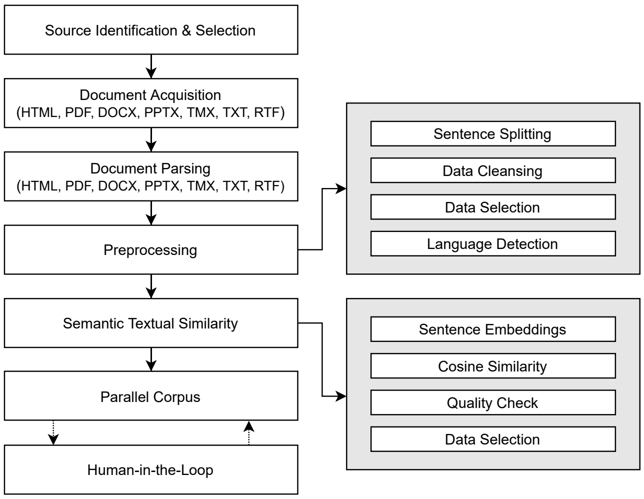


Fig. 4. Own corpus construction process

analyzed, and examples are used to discuss what the strengths and weaknesses are. If the strengths prevail, the data sources are included and considered in the data preparation step (see Figure 3). It is essential in this step to communicate and give examples of why data sources are suitable and why not. It is also advisable to ask for alternative formats that are easier to process in the case of proprieties or complex data formats. For example, the underlying DOCX files are often still available for PDF files.

Documents from accessed data sources are further processed in the “Data Preparation” step. This is a primarily automated step, which is discussed in Section III-B. In some cases, with high-quality datasets, these are of outstanding quality, so they can be adopted directly. In other cases, a “review” is necessary, since, e.g., faulty pagination due to OCR reduces the quality. These errors are quickly corrected but require manual reworking (cf. Section III-B0d). In addition, the evaluation process may also reveal that the resulting sentence pairs do not meet the requirements. If adjusting the settings (e.g., Alignment Score, Language Detection Confidence) cannot improve the quality, the dataset must be discarded. Here, it is essential to communicate the procedure and reasons to pay attention to any deficiencies in other data sources.

B. Automate repetitive, time-consuming tasks

Finding, extracting, preparing, and matching appropriate sentences across thousands of documents in numerous languages requires software assistance. In the following, we explain our approach (see Figure 4), using the processing steps from Koehn (2005) as mentioned in Section II.

a) Obtain & Extract: The origin of the data is diverse. In our real-world example, the information is located in two Content Management Systems (CMS), two Translation Management Systems (TMS), and a Product Database (PIM). There are also more than 5,000 PDFs containing general promotional material and stakeholder information. These PDFs can only be obtained from the company’s website via web crawling (We use the free command-line tool *wget* to retrieve files via HTTP). The *Document Acquisition* step utilizes various Application Programming Interfaces (APIs), file operations, and database queries (e.g., MySQL) to get as many documents as possible. Often, this step is a company-specific adaptation of our standard processes since the existing systems always have peculiarities (VPN accesses, access restrictions, API modifications, limitations). The raw data varies strongly in format and structure. On the one hand, documents are available as plain text, standardized XML (this includes DOCX, TMX), or in Excel spreadsheets. On the other hand, documents can also appear as PDF files and InDesign files, which have to be converted to HTML in the *Document Parsing* step first.

b) Split & Prepare: The motivation behind the *Preprocessing* step is to split texts into sentences (Sentence splitting is done with *sentence-splitter*, v1.4) and detect the language (We use *langdetect*, v1.0.9 for language detection). It also removes sentences that contain incorrect characters, are incorrectly encoded, or consist of less than three or more than 50 words. Sentences in languages that are not needed for the translation system are also removed. As a result, sentences are stored in individual files per language.

c) Map: This step ensures similarity on the one hand but also checks how similar the sentences are in terms of distance in characters and words. The alignment of semantically highly similar sentences is done by using SentenceTransformers [15] together with the LaBSE model. The LaBSE model supports 109 languages and works well for finding translation pairs in multiple languages. As a result, semantically identical sentences are expected here, which exist in two different languages. Since errors in language detection can occur in a few cases, character distance and cosine distance are used to ensure that these are not identical sentences of one language that differ only in a few words or characters (e.g., OCR errors). If the semantic similarity is very high (90%+) and distance in characters and words is given, a very good match is assumed, and the sentence pair is taken over into the parallel corpus.

d) Dataset Evaluation & Human-in-the-Loop: In cases where the data quality is supposedly not good enough, or inadequacies are detected automatically, a review process takes place. In this process, the sentence pairs are presented to a user for correction and approval (We use the Prodigy tool in this context). Users can edit both sentences and provide a comment, which can be helpful for further editing of the dataset. Once a sentence pair has been corrected, it is added to the training set and considered when the translation models are trained again. Furthermore, users of the translation software can specify that translations should be transferred to the review process if they are dissatisfied with the translation. In this way, incorrect translations are also corrected and taken into account in the next training of the models.

IV. Discussion

We believe that many companies have enough data to train translation solutions. One challenge is to access this data within a company. As shown, we use appropriate processes and technical tools for this purpose.

However, it is essential to understand and communicate that, ideally, this is a continuous process within the company. Language is subject to constant change, and domain-specific language changes and expands. It is necessary to continually train language models and also to constantly include new multilingual documents in this retraining process. Here, care must be taken to ensure that the rights to the documents remain with the company. Often, the idea is to crawl competitor websites or to use third-party product manuals as a data source. For legal reasons, we advise against this approach. We also check whether documents made available within the company meet the legal requirements for use.

A crucial role in improving the quality of in-house translation models is to involve employees in this process via the review process. We have had good experience enabling employees to report and correct “bad” translations. This improves the language model for everyone in the company and creates a sense of engagement and participation.

V. Conclusion

When it comes to self-hosted translation solutions tailored to a company’s language, the question of suitable training data quickly arises. In this paper, we have shown how we support companies in finding relevant datasets and preparing them so that they can be used for fine-tuning translation models. When working with companies, it often becomes apparent that valuable data is spread across various platforms, cloud storage, and systems and that there is rarely an overview of the data. Furthermore, the information is available in various file formats, some of which must be converted into text.

Furthermore, there is often no understanding of how data can contribute to a good translation system. For this reason, we have developed processes for identifying and preparing the relevant data in the company and using it to train the translation models. Furthermore, we have established evaluation and data preparation loops in terms of a human-in-the-loop approach that help to keep data quality high. We want to build on this approach with our future activities by establishing the data preparation and training process as a continuous process, thus enabling companies to develop their translation models continuously. In this context, we are already working with major partners from research and industry.

References

[1] M. Druskoczi, “Final report on the SME panel consultation on eTranslation and language technologies,” European Commission, Tech. Rep., 2020.

[2] M. Junczys-Dowmunt and et al., “Marian: Fast Neural Machine Translation in C++,” pp. 1–6, 2018.

[3] A. F. Aji, N. Bogoychev, K. Heafield, and R. Sennrich, “In Neural Machine Translation, What Does Transfer Learning Transfer?” in *Proceedings of the 58th Annual Meeting of the ACL.* ACL, 2020, pp. 7701–7710.

[4] L. Tian and et al., “UM-Corpus: A Large English-Chinese Parallel Corpus for Statistical Machine Translation.” in *LREC.* European Language Resources Association (ELRA), 2014, pp. 1837–1842.

[5] P. Koehn, “Europarl: A Parallel Corpus for Statistical Machine Translation,” in *MT summit*, vol. 5, Citeseer. ACL, 2005, pp. 79–86.

[6] M. Ziemski, M. Junczys-Dowmunt, and B. Pouliquen, “The United Nations Parallel Corpus v1.0,” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16).* ACL, 2016, pp. 3530–3534.

[7] M. Morishita, J. Suzuki, and M. Nagata, “JParaCrawl: A large scale web-based English-Japanese parallel corpus,” *arXiv preprint arXiv:1911.10668*, pp. 3603–3609, 2019.

[8] H. Schwenk, G. Wenzek, S. Edunov, E. Grave, and A. Joulin, “CCMatrix: Mining Billions of High-Quality Parallel Sentences on the WEB,” 2019. [Online]. Available: https://arxiv.org/abs/1911.04944

[9] H. Schwenk, V. Chaudhary, S. Sun, H. Gong, and F. Guzmán, “WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia,” *arXiv preprint arXiv:1907.05791*, pp. 1351–1361, 2019.

[10] C. Christodouloupoulos and M. Steedman, “A massively parallel corpus: the Bible in 100 languages,” *Language resources and evaluation*, vol. 49, no. 2, pp. 375–395, 2015.

[11] Y. Zhang, K. Wu, J. Gao, and P. Vines, “Automatic Acquisition of Chinese–English Parallel Corpus from the Web ,” in *European Conference on Information Retrieval.* Springer, 2006, pp. 420–431.

[12] F. J. Och, C. Tillmann, and H. Ney, “Improved Alignment Models for Statistical Machine Translation,” in *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.* ACL, 1999, pp. 20–28.

[13] N. Reimers and I. Gurevych, “Making monolingual sentence embeddings multilingual using knowledge distillation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing.* ACL, 11 2020, pp. 1–14, accessed: 01-04-2022. [Online]. Available: https://arxiv.org/abs/2004.09813

[14] R. Wirth and J. Hipp, “CRISP-DM: Towards a standard process model for data mining,” in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, vol. 1. Springer-Verlag London, UK, 2000, pp. 29–39.

[15] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing.* ACL, 11 2019, pp. 1–11, accessed: 01-04-2022. [Online]. Available: http://arxiv.org/abs/1908.10084

Implicit Statements in Healthcare Reviews: A Challenge for Sentiment Analysis

Joschka Kersting
Paderborn University
Paderborn, Germany
joschka.kersting@uni-paderborn.de

Frederik S. Bäumer
Bielefeld University of Applied Sciences
Bielefeld, Germany
frederik.baeumer@fh-bielefeld.de

***Abstract*—This paper aims at discussing past limitations set in sentiment analysis research regarding explicit and implicit mentions of opinions. Previous studies have regularly neglected this question in favor of methodical research on standard-datasets. Furthermore, they were limited to linguistically less-diverse domains, such as commercial product reviews. We face this issue by annotating a German-language physician review dataset that contains numerous implicit, long, and complex statements that indicate aspect ratings, such as the physician's friendliness. We discuss the nature of implicit statements and present various samples to illustrate the challenge described.**

***Index Terms*—Sentiment analysis; Natural language processing; Aspect phrase extraction.**

I. Introduction

Natural Language Processing (NLP) is a prominent subdomain of data science that is concerned with automatic processing of text data [1]. Natural language data is a challenge for machines because it is unstructured and contains imprecision, ambiguity, and vagueness [2]. There are characteristics that make standard language an efficient tool in human communication, but at which machine language processing regularly reaches its limits [2], [3]. A thriving topic in NLP and data science is Aspect-Based Sentiment Analysis (ABSA). ABSA "consists of two conceptual tasks, namely an aspect extraction and an aspect sentiment classification" [4], [5]. The aim is to categorize data by aspect and identify the sentiment polarity associated with each aspect. The subject of this analysis can be ratings of any kind, such as product ratings (e.g., cameras) or ratings of services (e.g., physician reviews). A well-known example is the battery of electronic products: While "*The battery of this phone is quite good*" is an explicit statement, '*'The phone lasts all day*" is the same statement but implicitly formulated [6].

As an example for a service review, earlier work [7] analyzes physician reviews and tries to apply a human-like language comprehension. The subject of physician reviews covers healthcare services that have been used by the author or a third party. These evaluative texts are published by users that describe their (dis-)satisfaction with a physician's treatment [8], [9]. A characteristic of these reviews is that they are often shaped by the sensitive physician-patient relationship. Due to this sensitive relationship, which should not be damaged despite review, many authors of reviews may resort to implicit statements in order to conceal the actual assessment somewhat. Implicit statements have the advantage that one does not have to commit oneself and can always deny in case of doubt. An example for implicit phrases is the following:

Example 1.
(1) "With this doctor, you don't just feel like a number."
(1) "Bei diesem Arzt fühlt man sich nicht nur als Nummer."

Example 1 shows that a patient was satisfied with the overall performance. The aspects "*time taken*" and "*friendliness*" are tangent to the positive statement. Both aspects are not explicitly mentioned but can be deduced. For a human reader, the connection arises from the overall context, since "*being a number*" is a phrase in German for feeling "*insignificant*" and "*unknown*". It can also be understood as "*to be treated without regard to personal circumstances*." While sentiment analysis approaches are quite capable of identifying the positive tenor, the domain-specific aspect classes remain unknown. To be able to process these reviews by machine with regard to aspects and associated sentiments, extensive datasets must be created and machine learning methods are trained with these datasets. A great number of previous research studies in this area focused on explicit statements and explicitly excluded implicit statements in some parts [10]. Moreover, studies often used the same datasets provided by Pontiki et al. [11]–[13], as survey papers demonstrate [14]–[16]. Hence, previous research is limited to what the datasets enable it to investigate. For example, the annotation guidelines of Pontiki et al. [17] state that only explicitly mentioned aspects should be annotated and that only one aspect in a sentence should be marked. Hence, researchers may train models that unlearn things that were not marked, due to these artificial boundaries. To make these statements visible by machine, we aim at implicit and explicit rating phrases in user-generated text. In this paper, we want to draw attention to the issue, show related work, and provide ideas to handle it.

This short paper is structured as follows: Section II presents related work for this paper. Based on this, we present examples for implicitness in review texts (Section III) and discuss them (Section IV). Finally, we conclude our work in Section V.

II. Related Work

In this section, we provide the related work with focus on deep learning for NLP (cf. Section II-A), sentiment analysis (cf. Section II-B), and user-generated content (cf. Section II-C).

A. Natural Language Processing

There has been great progress for NLP methods in recent years. Most notably, deep learning has evolved as the go-to method that improved the state-of-the-art in nearly all NLP tasks, such as question answering, sentiment analysis, and others. Here, transformers are the most important development [8], [18], [19]. Transformers apply a number of deep learning layers equipped with attention technology rather than recurrent neural networks [18]. This leads to favorable results and resource efficiency. Most notably, attention can process text sequences as a whole, i.e., it can weight words in a sentence according to their importance for the task it is learning [18], [19]. Recurrent networks such as Long Short-Term Memories (LSTMs) [20], on the other hand, process data sequentially, from the beginning to the end and hence only regard the part they have already seen [18], [19]. Furthermore, transformers have shifted the way neural networks are trained and handled for NLP. That is, large-scale models can be pre-trained on large amounts of raw text on a task that enables the construction of word vectors. These are representations of words, parts of words, or letters. The result is large models that can be shared with others. Industry practitioners can use these models or further train them for their specific data domain. Most notably, transformers are rather fine-tuned as a whole instead of inserting their vectors in other models. This process is called fine-tuning and describes a transformer receiving an additional layer and being trained on a downsstream task such as text classification for sentiment analysis. Here, all layers in the model are trained for this purpose [19], [21].

B. Sentiment Analysis

The second relevant area of research is sentiment analysis. We focus on ABSA in particular because we need to extract relevant statements from texts. Other works deal instead with document or sentence-based sentiment analyses such as full-text classification. ABSA can also be handled this way, but that is not purposeful, because the corresponding text spans must be extracted to enable further analyses for an in-depth knowledge of a text's contents and their explainability, [14]–[16]. This applies also to the distinction of implicit and explicit statements.

To describe ABSA and its components, we introduce the three sub-tasks here: ATE, ACC, and APC. ATE and ACC refer to Aspect Term Extraction and Aspect Class Classification [22]. These are usually conducted together [7] and describe the process of identification and categorization of aspect phrases in texts. APC refers to the Aspect Polarity Classification and describes the process of the sentiment polarity identification [22], e.g., negative or positive sentiment towards a cell phone battery. We need to conduct the first two steps ATE and ACC to extract implicit phrases from text. In contrast to studies such as Kersting & Geierhos [7], [8], we do not aim at topic-related extraction of aspect categories and their corresponding phrases. We deal with implicit and explicit phrases and their distinction. They set up aspect classes, extract them and further analyze them. However, the work also deals with implicit aspect phrases.

Several survey studies [14]–[16] present an overview of ABSA research. As can be seen, most works do not perform ATE, i.e., they do not provide in-depth analyses and go for the sentence or document-level. Moreover, they differ in the datasets they use [13], [23], [24]: Most studies use datasets from commercial review domains, e.g., for products or restaurants. They are not related to healthcare topics or physician reviews. Besides, most neural network approaches are based on rather common layer types such as (bi-)LSTMs or transformers. An example study for ABSA research is the one by De Clercq et al. [24]. They built an ABSA pipeline for social media data contents related to banking, retail, and human resources data. However, this does not deal with implicit statements. Garcia-Pablos et al. [25] present another example. They use topic modeling for finding thematic clusters. The topics found by topic modeling are not intuitive and cannot be clearly delimited for human users [26]. Hence, such approaches are very limited.

C. Physician Reviews

The third domain of relevant research includes physician reviews and research dealing with them. Such reviews serve as a sample domain for NLP research [7], but also are researched themselves [27], i.e., scholars want to investigate the content about healthcare providers and their performance. Physician reviews are published by users on Physician Review Websites (PRWs) sensibly and on the basis of trust [28]. They describe inter-personal issues and aspects such as the friendliness of a healthcare provider. This distinguishes them from commonly used commercial domains. Physician reviews serve as a good example for complex data domains [8]. On PRWs, there are two types of ratings: quantitative ratings such as stars or grades, and qualitative ratings such as texts. Both form a review. Quantitative ratings on PRWs are mostly positive [8] and there are numerous countries covered by PRW.

III. Implicitness in Aspect rich review texts

Having presented the most relevant research areas and selected studies from them, in this section, we present the challenge of implicit statements in service reviews. Here, we use data collected for earlier studies and related work [8], [29]. Hence, we have datasets of three different PRWs from three German-speaking countries. Here, we selected the review texts and split them into sentences using the NLP tool *Spacy*. We further applied some basic quality requirements such as a minimum and maximum length to avoid extremely long sentences spanning over a whole review without punctuation.

The annotation process was organized as follows: First, the sentences were randomized. We avoided annotator bias by not revealing further information about a review or the physician the sentence was written for. The annotation tool used is called *Prodigy*, a web-based tool that helps organizing and saving annotations consistently with multi-user support.

To keep the data quality high, we consistently monitored the annotations among team members. Edge cases were noted and talked through. However, the annotating persons were experienced with the task. We also wrote annotation guidelines where we distinguish explicit and implicit statements. This understanding was applied to the annotations. As a result, we have over 1400 sentences of which ca. 90% contain aspects. About 25% contain implicit aspect phrases and ca. 75% explicit. Despite our efforts, we cannot completely rule out subjectivity and are addressing this issue in future research.

Example 2.
(1) Well, the doctor is a nice person.
(1) Nun, der Doktor ist ein netter Mensch.
(2) When I meet him, he has always a warm smile in his face.
(2) Treffe ich ihn, hat er immer warmes Lächeln im Gesicht.

Sentence (1) in Example 2 does not state an aspect class explicitly, e.g., by saying "*The friendliness is positive.*" The rating towards the physician's friendliness would rather be expressed like in Sentence (2). But here again the questions arise whether naming a word that clearly hints to an aspect class, e.g., "*nice*" to "*friendliness*" is implicit or explicit. Besides, describing the "*warm smile*" is implicit: Here, no aspect class is indicated, but a human reader understands this interpersonal type of communication. As demonstrated, the distinction is not always clear or sharp. This is not uncommon in reviews of medical services, and we explain this with the sensitive doctor-patient relationship. Another explanation why reviewers so often resort to implicit statements may also be the strict rules of the PRWs, which protect against false reviews. Table I presents a number of examples for implicit and explicit aspect phrases as they persist in our dataset. As can be seen the distinction is challenging. To help readers understand our decisions, we accompany each sentence with an explanation. We applied a narrow understanding to implicit aspect phrases in contrast to previous research [7], [8]. They regard each case in which an aspect phrase is not directly mentioned as implicit and compare this to, e.g., Pontiki et al. [13], [17], who focus on directly named aspects. The following list demonstrates our comprehension of implicit and explicit aspect phrases:

Implicit phrases

- Statements apparent only when taken as a phrase or by taking the context as a whole into account, including idioms. An aspect class can be inferred from the phrase.
- Implicit phrases therefore do not contain explicit word choices (see underneath) from the aspect classes.

Explicit phrases

- At least one term of the known aspect classes is given, regardless of the inflectional form or part of speech; synonyms are included.
- It is made clear what is meant in the annotated phrase.

IV. Experiment and Discussion

In this short article, we want to draw particular attention to the need to give special consideration to implicit aspects in the evaluation of services and include relevant elucidating examples. But even the definition of these implicit aspects is not consistent. We have applied a narrow definition of implicit phrases, which makes the annotation task more challenging. Furthermore, the comprehension by Kersting and Geierhos [8] is easier to understand and apply due to its clear nature. However, we regard our approach as more sophisticated and hence useful for research, as easier understanding would limit the analyses. Kersting and Geierhos [8] do not research implicitness, but rather make use of implicit and explicit phrases regardless of whether they belong to either one of them. As an example, "*He is very friendly.*", taken from Table I, would be considered implicit by Kersting and Geierhos [8]. This is because the word "*friendly*" does not name the corresponding aspect phrase "*friendliness*", even though it (clearly) indicates it. However, our understanding is different: The example would be considered explicit, because the aspect class can be identified by human annotators.

Furthermore, to test whether computational models can learn this understanding, we trained several transformer models. These extract and classify implicit and explicit phrases from text based on our data. We conducted the experiment as a tagging task and thus handled two steps in one (ATE, ACC) and applied IO-tags, consistent with other works [8]. Based on Kersting and Geierhos [7], we applied XLM-RoBERTa [21] to the data, expecting to get favorable results. As the early experiment shows, it is possible to extract implicit phrases automatically, e.g., with an F1 score of 0.49 for implicit phrases. The overall accuracy for the model was 0.78, the F1 score 0.70. Words with explicit aspect mentions and irrelevant words using the O-tag were easier for the system to detect. This may be caused by fewer training data for implicit aspect phrases and their nature of being implicit while not having a limited or regularly occurring vocabulary.

Naturally, we conducted more and further experiments. A multi-label multi-class classification on the sentence-level succeeded and achieved good results. The Label ranking average precision (LRAP) [30] training XLM-RoBERTa large is 0.90. This is a very good score as it is close to 1, which would be best (between 0 and 1). Moreover, when training the tagger, other models such as BERT [19] in multiple variations (large, domain-trained, etc.) achieved almost the same scores like XLM-RoBERTa. A large German version of BERT [31] even outperformed XLM-RoBERTa large [21] achieving a macro F1 score of 0.74, an accuracy of 0.81 and an F1 score for implicit phrases of 0.52. Further used parameters were a small batch size (e.g., 4) and few epochs (8).

As can be seen, machine classification and extraction of implicit phrases is possible. However, we still need to further elaborate (and enrich) the definition of implicit aspect phrases. There is some related work in this area, but that is often focused on products and its elaboration does not fully reflect

TABLE I
EXAMPLES OF REVIEW PHRASES

Sentence	Class	Explanation
"*He is very friendly.*"	explicit	What is meant is explicitly made clear in the phrase.
"*He is not at all competent.*"	explicit	The competence is directly mentioned by an adjective.
"*He does not look me in the eye.*"	implicit	Only a human can guess that this is considered rude, because the issue is described in a phrase rather than a single adjective.
"*My wife and I have been going to this doctor for many years.*"	implicit	A trust relationship can only be derived from this statement.
"*I was treated immediately.*"	implicit	A human can guess that there was no waiting time.
"*Through him, a disease was finally detected.*"	implicit	Competence can be derived from the context.

the subtleties that come with the evaluation of services and other inter-personal situations [7, e.g.].

V. Conclusion

In this short paper, we have provided insight into implicit statements that occur in German physician reviews. Implicit statements are not a new phenomenon in machine processing of texts, especially not in sentiment analysis. However, implicit statements occur frequently in the review of medical services. This is also due to the sensitive physician-patient relationship. In principle, the challenge seems manageable with machine learning and clean, well-annotated datasets. We will continue to follow this path. the implications to the work of non-German structured languages. Finally, we know that this basic idea is not only valid for German, but that this challenge of implicit statements is also found in other languages. We are therefore eager to develop our approaches beyond German and in other domains. Furthermore, it can be a future research direction to investigate implicit phrases together with sentiment polarity, objectivity and subjectivity.

Acknowledgements

This work was partially supported by the German Research Foundation (DFG) within the Collaborative Research Center On-The-Fly Computing (SFB 901). We thank M. Cordes, and R. R. Mülfarth for their assistance with the data collection.

References

[1] I. Zeroual and A. Lakhouaja, "Data science in light of natural language processing: An overview," *Procedia Computer Science*, vol. 127, pp. 82–91, 2018.

[2] K.-U. Carstensen, C. Ebert, C. Ebert, S. Jekat, R. Klabunde, and H. Langer, Eds., *Computerlinguistik und Sprachtechnologie: Eine Einführung [Computational Linguistics and Language Technology: An Introduction]*, 3rd ed. Heidelberg, Germany: Spektrum Akademischer Verlag, 2010.

[3] H. Bußmann, *Lexikon der Sprachwissenschaft [Lexicon of Linguistics]*. Stuttgart: Alfred Kröner Verlag, 2008.

[4] Q. Jiang, L. Chen, R. Xu, X. Ao, and M. Yang, "A Challenge Dataset and Effective Models for Aspect-Based Sentiment Analysis," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: ACL, 2019, pp. 6280–6285.

[5] M. H. Phan and P. O. Ogunbona, "Modelling Context and Syntactical Features for Aspect-based Sentiment Analysis," in *Proceedings of the 58th Annual Meeting of the ACL*. Online: ACL, 2020, pp. 3211–3220.

[6] K. Schouten and F. Frasincar, "Finding implicit features in consumer reviews for sentiment analysis," in *International Conference on Web Engineering (ICWE)*. Toulouse, France: Springer, 2014, pp. 130–144.

[7] J. Kersting and M. Geierhos, "Human Language Comprehension in Aspect Phrase Extraction with Importance Weighting," in *Natural Language Processing and Information Systems*, ser. LNCS, E. Métais, F. Meziane, H. Horacek, and E. Kapetanios, Eds., vol. 12801. Saarbrücken, Germany: Springer Nature, 2021, pp. 231–242.

[8] ——, "Towards Aspect Extraction and Classification for Opinion Mining with Deep Sequence Networks," in *Natural Language Processing in Artificial Intelligence – NLPinAI 2020*, ser. Studies in Computational Intelligence (SCI), R. Loukanova, Ed. Cham, Switzerland: Springer, 2021, vol. 939, pp. 163–189.

[9] M. Emmert, F. Meier, F. Pisch, and U. Sander, "Physician Choice Making and Characteristics Associated With Using Physician-Rating Websites: Cross-Sectional Study," *Journal of Medical Internet Research*, vol. 15, no. 8, p. e187, 2013.

[10] M. Tubishat, N. Idris, and M. A. Abushariah, "Implicit aspect extraction in sentiment analysis: Review, taxonomy, opportunities, and open challenges," *Information Processing & Management*, vol. 54, no. 4, pp. 545–563, 2018.

[11] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 Task 4: Aspect Based Sentiment Analysis," in *Proceedings of the 8th International Workshop on Semantic Evaluation*. Dublin, Irland: ACL, 2014, pp. 27–35.

[12] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 Task 12: Aspect Based Sentiment Analysis," in *Proceedings of the 9th International Workshop on Semantic Evaluation*. Denver, CO, USA: ACL, 2015, pp. 486–495.

[13] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S. M. Jiménez-Zafra, and G. Eryiğit, "SemEval-2016 Task 5: Aspect Based Sentiment Analysis," in *Proceedings of the 10th International Workshop on Semantic Evaluation*. San Diego, CA, USA: ACL, 2016, pp. 19–30.

[14] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and Challenges of Aspect-based Sentiment Analysis: A Comprehensive Survey," *IEEE Transactions on Affective Computing*, 2020.

[15] J. Zhou, J. X. Huang, Q. Chen, Q. V. Hu, T. Wang, and L. He, "Deep Learning for Aspect-Level Sentiment Classification: Survey, Vision, and Challenges," *IEEE Access*, vol. 7, pp. 78 454–78 483, 2019.

[16] H. H. Do, P. Prasad, A. Maag, and A. Alsadoon, "Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review," *Expert Systems with Applications*, vol. 118, pp. 272–299, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418306456

[17] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval 2016 Task 5: Aspect Based Sentiment Analysis (ABSA-16) Annotation Guidelines," pp. 1–20, 2016.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," in *Proceedings of the 31st Conference on Neural Information Processing Systems*. Long Beach, CA, USA: Curran Associates, 2017, pp. 5998–6008.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of NAACL-HLT 2019*. Minneapolis, MN, USA: ACL, 2019, pp. 4171–4186.

[20] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsu-

pervised Cross-lingual Representation Learning at Scale," in *Proceedings of the 58th Annual Meeting of the ACL.* Online: ACL, 2020, pp. 8440–8451.

[22] T. C. Chinsha and J. Shibily, "A Syntactic Approach for Aspect Based Opinion Mining," in *Proceedings of the 9th IEEE International Conference on Semantic Computing*. Anaheim, CA, USA: IEEE, 2015, pp. 24–31.

[23] A. López, A. Detz, N. Ratanawongsa, and U. Sarkar, "What Patients Say About Their Doctors Online: A Qualitative Content Analysis," *Journal of General Internal Medicine*, vol. 27, no. 6, pp. 685–692, 2012.

[24] O. De Clercq, E. Lefever, G. Jacobs, T. Carpels, and V. Hoste, "Towards an Integrated Pipeline for Aspect-based Sentiment Analysis in Various Domains," in *Proceedings of the 8th ACL Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Copenhagen, Denmark: ACL, 2017, pp. 136–142.

[25] A. Garcia-Pablos, M. Cuadros, and G. Rigau, "W2VLDA: Almost Unsupervised System for Aspect Based Sentiment Analysis," *Expert Systems with Applications*, vol. 91, pp. 127–137, 2018.

[26] A. Mukherjee and B. Liu, "Aspect Extraction through Semi-Supervised Modeling," in *Proceedings of the 50th Annual Meeting of the ACL*, vol. 1. Jeju, South Korea: ACL, 2012, pp. 339–348.

[27] M. Emmert, U. Sander, A. S. Esslinger, M. Maryschok, and O. Schöffski, "Public Reporting in Germany: the Content of Physician Rating Websites," *Methods of Information in Medicine*, vol. 51, no. 2, pp. 112–120, 2012.

[28] J. Kersting, F. Bäumer, and M. Geierhos, "In Reviews We Trust: But Should We? Experiences with Physician Review Websites," in *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security (IoTBDS).* Heraklion, Greece: SCITEPRESS, 2019, pp. 147–155.

[29] J. Kersting and M. Geierhos, "Aspect Phrase Extraction in Sentiment Analysis with Deep Learning," in *Proceedings of the 12th International Conference on Agents and Artificial Intelligence: Special Session on Natural Language Processing in Artificial Intelligence (ICAART - NLPinAI 2020).* Valetta, Malta: SCITEPRESS, 2020, pp. 391–400.

[30] Scikit-learn Developers, "sklearn.metrics.label_ranking_average_precision_score – scikit-learn 1.0.2 documentation," https://scikit-learn.org/stable/modules/generated/sklearn.metrics.label_ranking_average_precision_score.html, 2020, accessed 04.04.2022.

[31] B. Chan, S. Schweter, and T. Möller, "deepset/gbert-large - hugging face," https://huggingface.co/deepset/gbert-large, 2022, accessed 2022-04-04.

Concept of an Inference Procedure for Fault Detection in Production Planning

Jan Michael Spoor
Team Digital Factory Sindelfingen
Mercedes-Benz Group AG
Sindelfingen, Germany
jan_michael.spoor@mercedes-benz.com

Jens Weber
Faculty of Business and Economics
University of Applied Sciences Zwickau
Zwickau, Germany
jens.weber@fh-zwickau.de

Simon Hagemann
Team Digital Factory Sindelfingen
Mercedes-Benz Group AG
Sindelfingen, Germany
simon.hagemann@mercedes-benz.com

Frederik S. Bäumer
Faculty of Business
Bielefeld University of Applied Sciences
Bielefeld, Germany
frederik.baeumer@fh-bielefeld.de

***Abstract*—To date, no implemented solution in manufacturing, i.e., in automotive industry, exists to support production planning with insights from production. A structured feedback loop from operations to planning is required to further improve production planning. This contribution discusses the limitations of an existing concept for an inference procedure from operations to new planning tasks using the findings from previous implementation studies. Using the constraints found in these studies, six principles for inference procedures are derived. Thus, the existing concept is renewed and a structured and specific approach in providing an inference procedure for planning activities of similar manufacturing systems is proposed. This approach is split into the different sub-tasks of data acquisition, fault detection, knowledge representation, and knowledge inference. Each sub-task has its unique state-of-the-art solutions, challenges, and limitations which have to be examined during further implementations. Most notably, the concept requires a definition of a normal model to derive fault events and error patterns, an embedding of the fault events in an ontology to create a knowledge base, and the definition of a metric to measure similarity between the current configuration in operation and new configurations of the production planning.**

***Index Terms*—Production Planning, Fault Detection, Knowledge Engineering, Data Mining, Case-Based Reasoning**

I. Introduction

The automation of production has a long tradition in the automotive industry. This industry has therefore been leading in the field of automation. Since the first initiatives of Henry Ford in the early 20th century, the assembly stages have especially been in the focus of automation [1]. Assembly processes require different levels of flexibility and in turn have led to different automation levels. The final assembly in automotive industry is still characterized by a high proportion of manual processes, while the Body-In-White (BIW) assembly is nowadays considered fully automated. The major step towards automation of production has been achieved as part of the third industrial revolution and is based on the use of computers, robotics, and electronics.

However, the ongoing fourth industrial revolution, which in Germany is considered as Industry 4.0, is based on increasing system connectivity and therefore the use of Cyber-Physical Systems (CPS) [2]. Cyber-Physical Production Systems are defined as "systems that integrate computation and physical processes [...]". The use of CPS continuously generates large amounts of sensor data. In automotive factories several terabytes of raw data are collected on a daily basis. However, the German high-tech strategy Industry 4.0 (I4.0) comprises more than the application of CPS. I4.0 targets the data continuity and autonomous orchestration of processes along the whole product-creation process [3].

In this contribution, we focus on a concept which assists the early phases of production planning in which the configuration of new assembly systems takes place. Therefore, real production data of operations is supposed to be a basis of the assistance system and requires the data continuity from the early phases of production configuration to the operational production. We conducted interviews with production planners in multiple European automotive Original Equipment Manufacturers (OEM), suppliers, and research institutes which have shown that there is no digital feedback loop from production back to the production planning.

In general, the computer-assisted assembly system configuration has for long been a discipline which has not been sufficiently regarded by research and industry [4]. Recently, research projects have been tackling the automation or at least the assistance of the early planning phases. Hagemann and Stark [5] provide an algorithmic approach in which the configuration is processed fully automated. The approach uses combinatorial optimization algorithms and determines the best production system configuration with the aim of minimizing investment costs. Other authors, such as Michalos et al. [6] and Michels et al. [7], published automated approaches for the design of assembly lines. However, to the best of the authors' knowledge, there are as of today no implemented solutions which aim at assisting the production planner based on the usage of real production data.

The following paragraphs describe a novel concept tackling this research gap. This novel concept introduces four sequential steps in the inference procedure: data acquisition, fault detection, knowledge representation, and knowledge inference. Through this approach, a comprehensive analysis and feedback of faults from operations to production planning is enabled. The focus of our contribution is on the automotive industry, but the concept is discussed in a universal manner, enabling the use in manufacturing in general.

In section II, a preceding concept for an inference procedure is presented, and its limitations found by our conducted implementation studies are discussed and analyzed. Thereupon in section III, the determined limitations are used to derive relevant principles, which must be taken into account when developing or implementing future concepts of inference procedures for fault detection. Subsequently, these principles are further aggregated to derive a mathematical problem description. Concluding in section IV, applying the problem description and principles, a novel concept for inference procedures using the four sequential steps is introduced. Furthermore, the state-of-the-art solutions and methods for each step of the introduced concept are discussed by conducting a comprehensive review of the methods in the current literature.

II. Derivations from the Preceding Concept

This contribution enhances the first concept by Gelwer et al. [8] and attempts to improve the approach by addressing limitations along the stages of the concept.

A. Description of Preceding Concept

The concept by Gelwer et al. [8] is a specific model to target a real manufacturing application at Mercedes-Benz Group AG (former Mercedes-Benz AG). The concepts sketch is given in Figure 1. The approach by Gelwer et al. [8] is summarized by following three stages:

(i) Identification of faults in the manufacturing production system during operations, the knowledge creation.
(ii) Set-up of a knowledge base.
(iii) Feedback of faults by identifying similar manufacturing systems in planning, an inference process applied in production planning procedures.

Stage (i) uses two approaches to identify faults. Firstly, an anomaly detection is conducted using data from Internet of Things (IoT) devices provided by a Manufacturing Service Bus (MSB), see Minguez [10], i.e., real-time data from devices, processes, and conditions. Secondly, Natural Language Processing (NLP) is applied for analyzing the error documentation. Input of the NLP can be plain text for a classification of errors, e.g., documented in an Enterprise Resource Planning (ERP) system or other third party sources. Described faults within shift logs should then be classified using standardized error codes. If both error detection and classification approaches work successfully, the results correspond, since documented errors by maintenance workers should also be visible in the data, and *vice versa* the detected anomalies should appear in the necessary documentation of the shift log.

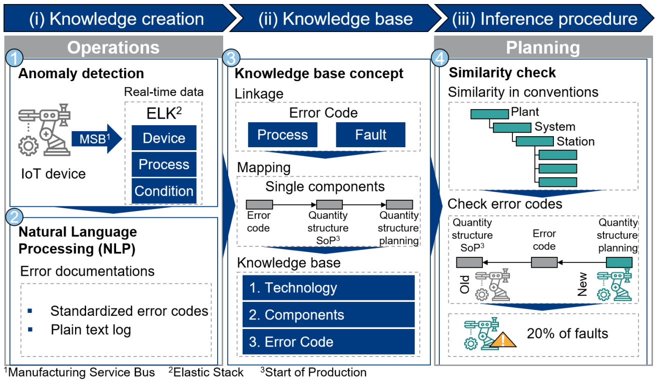


Fig. 1. Concept for data consistency checks between operation and production planning enabling an improved knowledge of past errors in planning by Gelwer et al. [8].

Stage (ii) is the set-up of the knowledge base by linking the corresponding technical description of the occurred faults and the affected processes within the error codes. These linked error codes are mapped with the hierarchical quantity structure of the manufacturing system after start of production and also in the stages of production planning, i.e., within the used library on the single component level. Further contextual information about the errors are stored, i.e., used technologies and parts numbers. The error and the additional contextual information are documented within the knowledge base.

Using the knowledge model, stage (iii) conducts an inference procedure in the case of a production planning process of a new manufacturing system. The new defined quantity structure in production planning is compared to the documented faults occurred in similar quantity structures after start of production within the library of stage (ii). Using the documented error code within the context of the quantity structure enables enriched information about possible problems or faults with the suggested component of the new planned manufacturing system. The proposed comparison using documented anomalies and faults of the past should be applied to the part and component level.

The concept by Gelwer et al. [8] was tested after its publication and further evaluated within the organizational structure of the Mercedes-Benz Group AG. By this practical application multiple limitations were detected, resulting in the necessity of a renewal of the approach. The limitations are ordered by corresponding stages not by importance.

B. Limitations of the Knowledge Creation

In stage (i), it is concluded that no so-called jack-of-all-trades algorithm or method for a consistent anomaly detection exists. This is not surprising since the free-lunch theorem implies that, considering all possible data, different anomalies, and targets of the detection process itself, no single algorithm is expected to solve all tasks [11]. This problem is directly relevant for the Mercedes-Benz Group AG and assumed for most organizations within manufacturing and automotive industry, as a wide variety of data types are available and in use. Using the proposed framework by Foorthuis [12], out of the 9 types with 63 subtypes of anomalies, 38 different subtypes from all 9 types of anomalies are expected within the data of the Mercedes-Benz Group AG. Although theoretically feasible, within an efficient organization or a framework of a business case individual implementations of all solutions per subtype are difficult to achieve. During the tests performed, the used algorithms, i.e., Isolation Forest, Multi-Layer-Perception, and K-Means, heavily relied on well-labeled data, test datasets, or required an extensive amount of prior investigation for setting up valid parameters. Thus, more generic approaches need to be defined, lessening the requirements for anomaly detection.

While the amount of data accessible is large enough, the error states are only occasionally and not consistently labeled. Furthermore, errors are quite rare. We estimate more than an additional decade of runtime using same configurations, as comparability is necessary, for creating sufficient error instances at the facilities of Mercedes-Benz Group AG. Many data of so called normal states [13] of the manufacturing system model exist, but real incidents of errors are rarely found in the available dataset or are often not documented enough in a consistent manner to draw structured conclusions.

Furthermore, the requirement to use real-time streaming data should be dropped since the proposed usage of real-time streaming data is technically complicated to implement [14]. More importantly, for a planning procedure with prior analysis of past data, real-time information processing is not necessary since no acute, short-term, and quick call for action is given.

The problem of stage (i) is exacerbated in the area of NLP. Only a limited amount of shift log entries exist, and from these only a limited number are related to specific errors. It is assumed that a higher training dataset size is highly beneficial for increasing the accuracy of NLP [15]. Furthermore, the documentation of manufacturing workers filling the shift books often lacks the required details in delimitation of the different types of faults or error codes. The insufficient documentation can be attributed to implicit knowledge of the workers, which is not known to the NLP algorithm. One current shortcoming of NLP and the development of artificial intelligence in general is the inclusion of implicit knowledge and human 'common sense' [16]. Therefore, shift logs could be used to determine if an error occurred but not what error occurred. Also, this still requires a larger amount of shift logs since the current entries are still too few to perform analysis.

To summarize, in stage (i) the classification of errors is a challenge of the concept and is not solvable with the current state-of-the-art tools proposed by Gelwer et al. [8].

C. Limitations of the Knowledge Base

While the linkage of the error code to the process is an important step in setting up and understanding the context of error messages, it is often not sufficient for later inferences. This procedure might correctly identify critical combinations of components and processes within the planning process, which lead to the described error states, but offers no sufficient information about the cause of the error that occurred and does not enable countermeasures except to dispense with the combination of component and process. Since faults are often foreshadowed by certain patterns and comparable faults can occur in different processes, linking these patterns might help to identify the specific error more precisely. This linkage enables a comparison it with similar faults, a comparison of solutions for these similar faults, and in conclusion enables targeted countermeasures. Therefore, the context of usage might also be an important factor in comparing the error with other occurring faults and their corresponding solutions. This enables a more detailed measure of criticality of the error and guides a decision on how to handle the error, if cost efficient, instead of avoiding it.

In addition, if the patterns are transferred and reused in stage (i), this additional context becomes an important part of the error classification and important to document within the error messages. Underlying faults, i.e., currently researched unwanted cold welding processes in holding pins, might be detectable by an overall pattern in the data not only by single faults and error messages. The feedback and usage of fault patterns is a useful addition to the knowledge base.

A helpful approach in stage (ii) is to identify affected components within their position in start of production as well as in the production planning libraries. The differentiation between start of production and planning might often be important since position, usage, and linked processes are changing during the production planning process in a manner that renders the reasoning behind the choice unclear. Nevertheless, using only the structural context of a resource within the quantity structure offers little information about the component and its use. Important contextual information is not documented within the quantity structure during production planning and start of production. A component might cause comparable errors within different quantity structures and contextual information about technologies, parts, usage, processes, and products might offer more explanatory value in describing errors.

D. Limitations of the Inference Process

This missing context within the quantity structure is even more important in setting up a similarity check in the inference process. The quantity structure itself, even if tracked within start of production and production planning, is not enough to detect similar set-ups. Very different quantity structures share comparable faults, and solving the faults in these different quantity structures might offer very important insights and enable solutions. While the quantity structure is certainly a part of the similarity measure, it must be enriched with more context. Similar quantity structures might behave very differently, and *vice versa* different quantity structures might be more comparable regarding documentation and detection of faults. Therefore, a fleshed-out ontology is needed to provide additional information about types, linkages, relations, and the interaction of product, process, and resources planned and deployed in this structure.

Furthermore, it is unclear how the similarity measure is set up since the quantity structure alone offers too little information. Even if the quantity structure is quite similar, it is difficult and unclear how to transform this similarity into a quantified measurement. Therefore, the proposed ontology must also offer the possibility to apply a quantifiable similarity measure. Based on the quantified measure, more similar setups and their respective faults should be given more weight when the planner is informed of potential errors by the inference process. Faults, risks, and solutions should be weighted by similarity. Therefore, the similarity, based on a metric quantifying the distance between the past and new planned configuration, becomes a measure on how likely a similar fault, which occurred prior in the compared past configuration, will occur in the new configuration. The predicted error-proneness of the new configuration is assumed to be correlated to the distance measure between the new and past configuration and the prior measured error-proneness of the past configuration. This metric needs to be developed and embedded within the proposed ontology.

III. Derived Requirements and Problem Definition

Considering the discussed limitations, relevant principles for future concepts can be derived and a mathematical problem formulation can be set up.

A. Requirements for Future Concepts

The relevant findings from the discussion of the preceding concept can be expressed by the following six principles:

1) Since faults are rare in the data, an approach using labeled faults requires more labeled training data for a valid classification of errors than currently available. As an alternative, a normal model needs to be defined, and all data deviating from the normal model should be classified as generic faults. The use of only supervised approaches is not recommended.
2) Since shift logs contain no information about the exact errors but can be used to identify if any error occurred, they enable spotting of time frames of interest for finding error patterns. Not all data are analyzed but data occurring during days with entries in the shift logs are.
3) Using the deviations from the normal data, these findings can then be compared regarding their unique patterns and segmented for building a new fault classification structure. The classified patterns are then the classification criteria for all anomalies.
4) As these fault patterns might be highly individual for each configuration, the configurations need to be described in a more meaningful way. A simple description within the quantity structure of production planning is not sufficient. Each configuration must be enriched with contextual data which then enables a deeper contextual anomaly detection and a real causality analysis.
5) Furthermore, because configurations are solely dependent on their quantity structure, an additional ontology must be created to make configurations more specific and comparable beyond the quantity structure.
6) Using this ontology, a metric must be developed, capable of comparing the similarity of configurations independently of their hierarchical position within the quantity structure of production planning and start of production.

If the proposed principles are considered, a risk assessment of a new planned configuration can be conducted by a comparison with the past configurations. By applying a similarity score based on a metric using ontologies and combining this information with the risk of a fault event, the risk of the observed new configuration in the production planning process is derived. This problem description and the resulting approach is related to case-based reasoning [9].

B. Mathematical Problem Formulation

To address the requirements discussed, we build a fundamental logic on how to feed errors back.

First, each resource is assumed to have a certain and known configuration θ_k out of a finite set of all possible configurations for these resource types. The configuration depends strongly on the Products, Processes, and Resource (PPR) model.

$$\theta_k \in \Theta = \{\theta_1, ..., \theta_K\} \tag{1}$$

Each resource and its specific configuration k have a finite set of possible faults or error states. Each error state j is defined as follows:

$$e_j \in E_k = \{e_1, ..., e_J\} \tag{2}$$

For each error j in configuration k there exists a certain probability $r_{j,k}$ that the error occurs.

$$r_{j,k} = P(e_j \mid \theta_k) \tag{3}$$

The risk of any error occurring in configuration k is then given as following expression:

$$r_k = \sum_{e_j \in E_k} P(e_j \mid \theta_k) \tag{4}$$

If each risk does not contribute equally to the perceived economic risk, a weight w_j of each error can be applied.

If additionally a second configuration k^* exists, there exists an amount of errors which are both present in the configuration k and k^*.

$$E_{k \cap k^*} \in E_k \cap E_{k^*} \neq \emptyset \tag{5}$$

The risk of any error occurring in configuration k is analogously given by following expression:

$$r_{k^*} = \sum_{e_j \in E_{k \cap k^*}} P(e_j \mid \theta_{k^*}) + \sum_{e_j \notin E_{k \cap k^*}} P(e_j \mid \theta_{k^*}) \tag{6}$$

If configuration k^* is not within operation and currently just a configuration during production planning, no estimation of $P(e_j \mid \theta_{k^*})$ can be conducted. But if configuration k^* and k are similar enough, it is assumed that the set of errors within

ISBN: 978-1-61208-953-9

configuration k^* but not within configuration k is very small. The occurrence of completely new errors is unlikely.

$$\sum_{e_j \notin E_{k\cap k^*}} P(e_j \mid \theta_{k^*}) \approx 0 \tag{7}$$

To conduct a valid inference procedure, a metric defining a distance measure $\Delta(\theta_k, \theta_{k^*})$ is necessary to compare k and k^*. This metric should then give an approximation of the possible error states using the configuration k as base.

$$r_{k^*} \approx \sum_{e_j \in E_{k\cap k^*}} P(e_j \mid \theta_{k^*}) \approx \sum_{e_j \in E_k} P(e_j \mid \Delta(\theta_k, \theta_{k^*}), \theta_k) \tag{8}$$

For each error, a relation between configuration k and k^* dependent on the distance measure between them is assumed.

$$P(e_j \mid \Delta(\theta_k, \theta_{k^*}), \theta_k) \sim P(e_j \mid \theta_k) \circ \Delta(\theta_k, \theta_{k^*}) \tag{9}$$

If this relation is measurable by, e.g., a correlation analysis, the inference is then a useful risk measure for the error-proneness of configuration k^*. Therefore, in order to conduct a risk assessment of a new configuration k^*, the following challenges need to be addressed:

1) The risk assessment of base configuration k is necessary.
2) There needs to be a valid definition of a metric $\Delta(\theta_k, \theta_{k^*})$.
3) Using the metric and risk assessment of k, a risk assessment of k^* must be derived.

IV. Proposed Concept of Inference Procedure

Based on the existing concept and the derived principles and mathematical problem formulation, a new concept is developed and presented in this section. This concept is then discussed along its proposed steps.

A. Concept Overview

First, the concept overview and core ideas are presented. Since the proposed concept needs to include additional information about ontologies, a relevant comparable fault diagnostics method is proposed by Zhou et al. [17]. The structural set-up of the proposed concept by Zhou et al. [17] is included into the proposal by Gelwer et al. [8] and adapted to meet all defined principles. Our proposed concept for an inference process is sketched in Figure 2.

The concept is split into four constructive steps and three input models:

The (1) data acquisition describes the process of collecting, processing, storing, and providing the data in order to conduct a fault detection.

The (2) fault detection accesses the normal model to use it as base for a detection if any kind of event happened and to describe the event patterns. The normal model utilizes the insights of principle 1) since more normal data are available and a normal model can be set up using training datasets. Also, this utilizes principle 2) by first detecting if any event happened before classifying or describing the event.

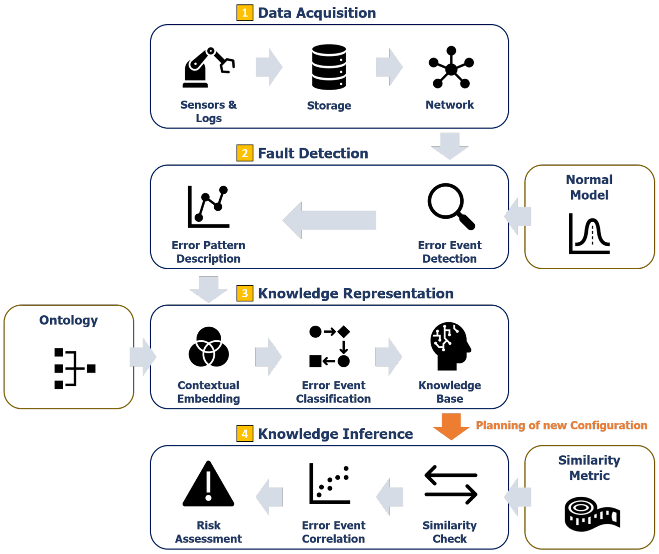


Fig. 2. Proposed concept for an inference process based on knowledge representation. The model is enabled by four constructive steps and three inputs: An ontology of the configurations, a similarity metric, and a normal model of the data.

In the (3) knowledge representation, the pattern and error events are then embedded in the ontology of the configuration, in which the event took place, and used for an event classification. A notable ontology in a similar use case is defined by Ming et al. [18] using, adapted to the discussed use case, components' taxonomy, properties, and relationships regarding features, operation, and quantity structure. This kind of ontology fulfills principle 5). If the event is classified by comparing the error pattern with similar events, as requested by principle 3), the knowledge base represents specific types of error events. These events are documented by well-defined patterns and are occurring within delimited areas and applications according to the ontology, as principle 4) suggests. For each error, the probability of occurrence $r_{j,k}$ can be determined by predictive pattern mining of the specific error event.

The (4) knowledge inference is then conducted when a new configuration is planned. The new planned configuration is compared to the configuration in the knowledge base by applying the defined metric as per principle 6). This similarity is the expression of the term $\Delta(\theta_k, \theta_{k^*})$ and is then used in the error correlation. The correlation process itself is represented by equation (9). This correlation is then used to calculate a risk assessment r_{k^*}.

The constructive steps and their challenges, current state-of-the-art solutions, requirements, and derived research questions are discussed in more detail in the following subsections.

B. Data Acquisition

The data acquisition is similar to the proposal by Gelwer et al. [8] and also proposes the usage of a MSB as described by Minguez [10]. The MSB uses a multitude of interfaces, which need an implementation within the overall IT infrastructure of a manufacturing company. The MSB acts therefore as a universal communication layer enabling the integration of all data from the shop floor for our proposed method [19]. The MSB solves the challenge of integrating a multitude of IoT data [1] and can be, as proposed by Gelwer et al. [8], transferred by Message Queuing Telemetry Transport (MQTT) protocols in JavaScript Object Notation (JSON) using process parameters and information from the Programmable Logic Controller (PLC) combined with informations from the ERP.

An additional part of the data acquisition is the provision of shift logs as possibility to detect error events. The application of an NLP is then necessary to detect if error events happened and mark these dates within the data for the error event detection research. The marked dates are primarily important within the error event detection of the fault detection step.

C. Fault Detection

While it is assumed that in most cases a supervised anomaly detection might offer more insights due to the incorporation of application-specific knowledge, the rare occurrence of faults within the application in production planning makes it difficult to create robust and generalized methods in this case [13].

Therefore, the fault detection process becomes for supervised methods a case of one-class variation since mostly normal data are available. A possible method would be the application of a one-class Support Vector Machine (SVM) as described by Schölkopf et al. [20]. Alternatively, unsupervised approaches can be used but risk the falsely positive detection of noise in the data as faults [13].

One challenge is the false inclusion of anomalies, despite being rare, into the normal data. An anomaly could easily be misclassified as normal. These rare instances could result in a more sensitive one-class SVM. A robust method for fault detection using one-class SVM is given by Yin et al. [21].

Another possible method to apply in these cases are Kernel Principal Component Analysis (PCA) for novelty detection. Kernel PCAs map the mostly normal instances containing training dataset into a feature space. The squared distance to the corresponding principal subspace is then a measure for anomalous data [22].

Different from an approach using one-class SVM or Kernel PCA methods, would be the full modeling of the normal machinery behavior to deduct anomalies by comparing the delta between prediction and measurement. These methods are more complex since they require a set-up of a complete normal model of the planned system but are very robust and rely on simpler distance-based comparisons between predictions and measurements. The quality assessment of such models is not the novelty detection method but the quality of the model itself. These models are achieved by a comprehensive Digital Twin defined as a simulation using physical models [23].

A possible model, which does not require a full Digital Twin, is an autoregressive time series AR(p) of an order p comparing the measurements with distance-based metrics, i.e., the Mahalanobis distance [24].

For rare instances of anomalies, the model might not be necessary to set up, but the state signal itself is the base of comparison, i.e., by comparing different windows of a signal. The cross correlation entropy between two windows can be measured and windows containing anomalies will result in a larger entropy of cross-correlation [25].

Further research is necessary to determine the best model or approach for the fault event detection since no clear recommendation for one specific approach is currently possible.

For a pattern identification of fault events, the rarity of faults must be taken into account. The pattern identification task becomes a problem of infrequent patterns. Therefore, descriptive tasks should be used to identify comprehensible patterns which are later labeled in the knowledge representation step.

Since it is expected that faults occur in very specific scenarios and are foreshadowed by co-occurrences prior or after the fault event, these co-occurrences can be used to describe the pattern and delimit it from other faults. Therefore, methods of descriptive association rule mining might be most useful in the pattern identification [26].

D. Knowledge Representation

Important input for the set up of the knowledge representation is the prior set-up of an ontology.

An ontology is defined as the model representing the semantics of the domain model. A knowledge graph is the result if data instances are acquired, integrated into an ontology, and additionally a reasoning is applied to derive new knowledge [27]. This differentiates an ontology and the resulting knowledge graph. The ontology itself offers little specific insight on the domain, but the domain becomes relevant when applied in the knowledge graph [28]. Therefore, the ontology is the input model and the application and contextual embedding of the detected faults integrated into the ontology becomes a knowledge graph which acts as the knowledge base for the following inference procedure.

The main challenge of a valid knowledge representation is the definition of a useful ontology, since for the presented application two obstacles are limiting the usage of currently available semantic model-based ontologies in Industry 4.0 applications as described by Yahya et al. [28]:

1) Production models do not fully follow the linked data principles and require a new vocabulary instead of the re-usage of current used vocabularies.
2) The scope of currently used ontologies is too application-specific and not applicable in all areas of the production.

A notable contribution in defining a possibly relevant ontology for contextual Industry 4.0. systems is given by Giustozzi et al. [29]. Besides the already defined necessary relationships (see subsection A), the ontology by Giustozzi et al. [29] uses dedicated resource, situation, process, time, location, and sensor ontologies.

Most relevant for production planning in the automotive industry are the domains of Product, Process, and Resources, bundled in the PPR concept [30]. A detailed exemplary set-up of the ontology of the PPR concept is given by Agyapong-Kodua et al. [31]. An applicable ontology for the proposed concept must therefore combine the aspects of the PPR concept as well as the aspects of ontologies for a contextual anomaly detection. Most beneficial would be a smooth integration into the existing PPR ontology models which are currently in use by manufactures and automotive.

The usage of more refined ontologies is enabled by the progressive efforts of companies in the holistic implementation of a Digital Twin, defined as a comprehensive physical but also functional description of components, products, and systems which enables insights in later lifecycle phases [32]. This also requires a Digital Twin definition as the sum of logically related data represented by semantic data models [33]. While the call for a more refined ontology seems to be a difficult requirement at first glance, it might be solved parenthetically due to the set-up of Digital Twins.

In the contextual embedding step of the detected faults, only the linked and semantic description of the faults are capable of setting up contextual error identifications, thus enabling a contextual error classification. By classifying the errors in the knowledge representation step, this enables a contextual anomaly detection using the error patterns and the domain knowledge. A notable exemplary method for sensor data is given by Hayes and Capretz [34] using cluster analysis to define sensor profiles and enabling a contextual analysis within the profiles. This application could be relevant for the discussed measurements of streaming sensor data in the data acquisition step. Also, the contextual anomaly detection might enable the previously unsuccessful NLP of shift logs for error classification. An exemplary application is described by Mahapatra et al. [35].

After the fault is described within the ontology and the fault patterns are delimited, the fault j is uniquely classified and combined with the corresponding pattern description which is the error event classification step. After classification, the past data are searched for the fault patterns. The found instances of faults of the same type within the specific configuration k are then counted. This pattern mining enables a calculation of the error-proneness probability $r_{j,k}$.

This task can be conducted by applying different predictive pattern mining algorithms [26]. Which algorithm for the prediction and classification of faults performs most usefully in the described use case must be further researched.

The error-proneness probability and error pattern embedded in the ontological description of the configuration build up the knowledge base.

E. Knowledge Inference

Since the concept is set within the idea of the digital factory, the product development and the production planning are parallelized. Therefore, the concept should also support a rough planning as the first step of the production planning process. Furthermore, in the applied planning within manufacturing companies, focus of the rough planning is often more on resources than processes since resources are main part of the cost calculation [36]. Within the rough planning a quantity structure and a 3D layout based on the used library containing the single components and parts of the production system is set up. Only in the detailed planning the supplier is involved, optimizing the quantity structure up until the start of production [37]. The higher the degree of maturity, the more information and usefulness a rough production planning provides [38].

Therefore, the similarity measure can only be as good as the rough production planning. Since more information is added during planning, more ontology types are also added and then enable better similarity measures. Conversely, this also means that the ontology must be imposed to planners, suppliers, and operation since a documented ontology in the production planning and start of production state benefits the significance of the analysis.

If the ontology is documented diligently, the main challenge is the set-up of a useful metric. Already in the definition of the metric a contrary objective arises: the metric must be set up in a way to properly describe the error-proneness of planned configurations k^* based on current configurations k, but the error-proneness of the planned configurations is itself derived from the distance measure of the metric. This conflict of goals makes an objective definition difficult. It is assumed that the metric to be defined is more likely a fuzzy similarity assignment, i.e., a probability that the configurations are similar, than a hard assignment listing the most similar configurations. This is the case, since similarity is often context-specific and different from equality measured in degrees [9]. Even then, a fuzzy assignment still needs to be quantified and must be tenable even under a generous error interval in real-world applications considering domain knowledge of the planners. The set-up of a metric is one of the current challenges and open research questions of the concept.

If a valid metric is defined, the correlation analysis following equation (9) is conducted to calculate a risk assignment of the new configuration k^* in production planning. Commonly-used tools in case-based reasoning include, e.g., regressions, bayesian learning, and Artificial Neural Networks, which might also be applicable in the presented use case [9].

V. Conclusion

Though purposefully improving the former concept proposed by Gelwer et al. [8] through the six principles established, our new proposed concept still has open challenges and needs further efforts to address these issues, namely the set-up of a valid ontology within the manufacturing system description and the derivation of a useful metric to determine similarity between configurations. Further challenges are the selection of a useful fault detection method and the set-up of a use case oriented pattern mining.

Nevertheless, within this contribution we were able to determine requirements of inference procedures and make a targeted proposal for future research in this area. In particular, the six defined principles and the proposed mathematical correlation definition between the error-proneness of planned configurations k^* based on current configurations k contribute to the current efforts towards building an inference procedure.

Furthermore, our proposed concept acknowledges the shortcomings of the former concept and proposes an advanced structure. Our proposal uses the stages of data acquisition, fault detection, knowledge representation, and knowledge inference. These stages are enabled by the definition of a normal model as a basis for fault detection, an ontology for a valid representation, and a similarity metric in order to be able to carry out target-oriented comparisons.

The authors plan to examine the proposed concept in more detail and implement use-case oriented applications of the concept in production planning in future studies.

Acknowledgment

We thank all colleagues and former colleagues who provided data, concepts and development support of the research work within Mercedes-Benz Group AG and the University of Applied Sciences Zwickau.

The Mercedes-Benz Group AG funds this research.

References

[1] T. Bauernhansl, M. Hompel, and B. Vogel-Heuser (eds), Industrie 4.0 in Produktion, Automatisierung und Logistik: Anwendung, Technologien, Migration. SpringerLink, Springer Vieweg, Wiesbaden, 2014.

[2] Plattform Industrie 4.0, Plattform Industrie 4.0 - Digital Transformation "Made in Germany". 2019.

[3] acatech - Deutsche Akademie der Technikwissenschaften, Key themes of Industrie 4.0. 2019.

[4] S. Hagemann and R. Stark, "Automated Body-in-White Production System Design: Data-Based Generation of Production System Configurations," In: Proceedings of the 4th International Conference on Frontiers of Educational Technologies, ACM, New York NY, pp. 192-196, 2018.

[5] S. Hagemann and R. Stark, "An optimal algorithm for the robotic assembly system design problem: An industrial case study," In: CIRP Journal of Manufacturing Science and Technology, vol. 31, pp. 500-513, 2020.

[6] G. Michalos, A. Fysikopoulos, S. Makris, D. Mourtzis, and G. Chryssolouris, "Multi criteria assembly line design and configuration - An automotive case study," In: CIRP Journal of Manufacturing Science and Technology, vol. 9, pp. 69-87, 2015.

[7] A. S. Michels, T. C. Lopes, C. G. Sikora, and L. Magatão, "The Robotic Assembly Line Design (RALD) Problem: Model and case studies with practical extensions," In: Computers and Industrial Engineering, vol. 120, pp. 320-333, 2018.

[8] E. Gelwer, J. Weber, and F. Bäumer, "A Concept of Enabling Data Consistency Checks Between Production and Production Planning Using AI," In: Proceedings of the 17th International Conference on Applied Computing, pp. 139-142, 2020.

[9] M. M. Richter and R. O. Weber, Case-Based Reasoning. Springer, Berlin, Heidelberg, 2013.

[10] J. Minguez, "Der Manufacturing Service Bus," In: E. Westkämper, D. Spath, C. Constantinescu, and J. Lentes (eds), Digitale Produktion. Springer, Berlin, Heidelberg, 2013.

[11] Y. Ho and D. Pepyne, "Simple Explanation of the No-Free-Lunch Theorem and Its Implications," In: Journal of Optimization Theory and Applications, vol. 115, pp. 549-570, 2002.

[12] R. Foorthuis, "On the Nature and Types of Anomalies: A Review of Deviations in Data," In: Int J Data Sci Anal, vol. 12, pp. 297-331, 2021.

[13] C. C. Aggarwal, Outlier Analysis. Springer Science+Business Media, New York, 2013.

[14] A. Lavin and S. Ahmad, "Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark," In: IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 38-44, 2015.

[15] M. Banko and E. Brill, "Mitigating the Paucity-of-Data Problem: Exploring the Effect of Training Corpus Size on Classifier Performance for Natural Language Processing," In: Proceedings of the first international conference on Human language technology research, pp. 1-5, 2001.

[16] N. J. Nilsson, The Quest for Artificial Intelligence. Cambridge University Press, 2009.

[17] Q. Zhou, P. Yan, H. Liu, and Y. Xin, "A hybrid fault diagnosis method for mechanical components based on ontology and signal analysis," In: J Intell Manuf, vol. 30, pp. 1693-1715, 2019.

[18] Z. Ming, C. Zeng, G. Wang, J. Hao, and Y. Yan, "Ontology-based module selection in the design of reconfigurable machine tools," In: J Intell Manuf, vol. 31, pp. 301-317, 2020.

[19] D. Schel, C. Henkel, D. Stock, O. Meyer, G. Rauhöft, P. Einberger, M. Stöhr, M. A. Daxer, and J. Seidelmann, "Manufacturing Service Bus: An Implementation," In: Procedia CIRP, vol. 67, pp. 179-184, 2018.

[20] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating Support of a High-Dimensional Distribution," In: Neural Comput, vol. 13, no. 7, pp. 1443-1471, 2001.

[21] S. Yin, X. Zhu, and C. Jing, "Fault detection based on a robust one class support vector machine," In: Neurocomputing, vol. 145, pp. 263-268, 2014.

[22] H. Hoffmann, "Kernel PCA for novelty detection," In: Pattern Recogn, vol. 40, no. 3, pp. 863-874, 2007.

[23] F. Tao, J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui, "Digital twin-driven product design, manufacturing and service with big data," In: Int J Adv Manuf Technol, vol. 94, pp. 3563-3576, 2018.

[24] M. Hau and H. Tong, "A practical method for outlier detection in autoregressive time series modelling," In: Stochastic Hydrol Hydraul, vol. 3, pp. 241-260, 1989.

[25] T. Wang, W. Cheng, J. Li, W. Wen, and H. Wang, "Anomaly detection for equipment condition via cross-correlation approximate entropy," In: MSIE 2011, pp. 52-55, 2011.

[26] S. Ventura and J. M. Luna, Supervised Descriptive Pattern Mining. Springer, Cham, 2018.

[27] L. Ehrlinger and W. Wöß, "Towards a Definition of Knowledge Graphs," In: SEMANTiCS, vol. 48, pp. 1-4, 2016.

[28] M. Yahya, J. G. Breslin, and M. I. Ali, "Semantic Web and Knowledge Graphs for Industry 4.0," In: Appl. Sci. 2021, vol.11, article 5110, 2021.

[29] F. Giustozzi, J. Saunier, and C. Zanni-Merk, "Context Modeling for Industry 4.0: an Ontology-Based Proposal," In: Procedia Computer Science, vol. 126, pp. 675-684, 2018.

[30] R. B. Ferrer, B. Achmad, D. Vera, A. Lobov, R. Harrison, and J. L. Martínez Lastra, "Product, process and resource model coupling for knowledge-driven assembly automation," In: Automatisierungstechnik, vol. 64, no. 3, pp. 231-243, 2016.

[31] K. Agyapong-Kodua, C. Haraszkó, and I. Németh, "Recipe-based Integrated Semantic Product, Process, Resource (PPR) Digital Modelling Methodology," In: Procedia CIRP, vol. 17, pp. 112-117, 2014.

[32] S. Boschert and R. Rosen, "Digital Twin - The Simulation Aspect," In: P. Hehenberger, D. Bradley (eds): Mechatronic Futures. Springer, Cham, 2016.

[33] M. Kunath and H. Winkler, "Integrating the Digital Twin of the manufacturing system into a decision support system for improving the order management process," In: Procedia CIRP, vol. 72, pp. 225-231, 2018.

[34] M. A. Hayes and M. A. M. Capretz, "Contextual anomaly detection framework for big sensor data," In: Journal of Big Data, vol. 2, article 2, 2015.

[35] A. Mahapatra, N. Srivastava, and J. Srivastava, "Contextual Anomaly Detection in Text Data," In: Algorithms, vol. 5, no. 4, pp. 469-489, 2012.

[36] S. Hagemann, A. Sünnetcioglu, and R. Stark, "Hybrid Artificial Intelligence System for the Design of Highly-Automated Production Systems," In: Procedia Manufacturing, vol. 28, pp. 160-166, 2019.

[37] D. Weidemann and R. Drath, "Einleitung," In: R. Draht (eds), Datenaustausch in der Anlagenplanung mit AutomationML. VDI-Buch. Springer, Berlin, Heidelberg, 2010.

[38] W. Walla, Standard- und Modulbasierte digitale Rohbauprozesskette. Frühzeitige Produktbeeinflussung bezüglich Produktionsanforderungen im Karosserierohbau der Automobilindustrie. Dissertation, Karlsruhe Institute of Technology, Karlsruhe, 2015.

Track Me If You Can: Insights into Profile Interlinking on Social Networks

Sergej Denisov
Bielefeld University of Applied Sciences
Bielefeld, Germany
sergej.denisov@fh-bielefeld.de

Frederik S. Bäumer
Bielefeld University of Applied Sciences
Bielefeld, Germany
frederik.baeumer@fh-bielefeld.de

Michaela Geierhos
Universität der Bundeswehr München
Munich, Germany
michaela.geierhos@unibw.de

***Abstract*—Social networks shape today's Web with modern communication capabilities and the ability to share heterogeneous media. The various networks have different focuses and reach different user groups so that users are often registered and active on multiple networks to take advantage of the respective benefits. In the ADRIAN project, we study threats to users on the Web, focusing on the creation of Digital Twins of real users. Here, we investigate the interlinking of user profiles on Social Networks and derive insights that help us model Digital Twins. We discuss the possibilities of using links to find additional profiles and assign them to users. To do this, both the links and the information in the profiles are examined. Only with high-quality data, it is possible to warn users reliably about the dangers of disclosing data.**

***Index Terms*—Social network services, Data privacy**

I. Introduction

People share their opinions and daily experiences on Online Social Networks (OSNs) [1]. However, platforms differ in terms of functionality and user experience. For this reason, users share their posts on the same topic on different OSNs (so-called cross-platform content sharing) [2]. On average, a Web user is expected to have 7.5 social media accounts in 2022 [3]. Moreover, the various OSNs often have different audiences, as well as different rules, therefore the posts are adapted accordingly by the users, especially in language style [2]. Significant differences were identified between the platforms in terms of usage and posting behavior. For example, Twitter is primarily used for information purposes, Twitter and Instagram for social sharing, and Instagram for entertainment purposes [4].

As a result, the available information of individual users also differs from one social network to another. By interlinking users across OSNs, very comprehensive user profiles can be obtained so that, on the one hand, their entire profile and behavior and, on the other hand, their preferences, activities, and friend network can be reconstructed. In the area of cybercrime the digital footprint can be used to track and target such users and create Digital Twins (DTs) [5]. To more effectively combat cyberbullying and identity theft, it is necessary to develop preventive methods that uncover the digital footprints, reveal the links across social media profiles, and thus point out the associated individual exposures.

The research project ADRIAN (*Authority-Dependent Risk Identification and Analysis in online Networks*) focuses on the disclosure of information by individuals in Web 2.0 [5]. This is not a new topic, but has been researched for some time now [6]. With the rise of modern OSNs, the threat to the author through published information became more concrete [7]. One form of this threat is doxing, which is the collection and publication of information about a person. Cyberbullying such as doxing is increasing [8], so automated solutions are being worked on to detect and prevent it [9]. From the user's point of view, it is often difficult to understand that information distributed across different "places on the Web" can, in combination, pose a threat. We attempt to make this threat visible to users by merging the information and modeling it as DT. However, merging information across OSNs (i.e., profile interlinking) is a data processing challenge and is the subject of current research [10].

In this short paper, we look at the underlying data available for creating link profiles and initializing DTs. For this purpose, we look at sample data from Twitter, YouTube, Facebook, and Instagram and highlight challenges that arise from a data science perspective. In this context, we focus on the possibilities of reliable profile merging and data quality. The structure of this paper is as follows: In Section II, related work is outlined. Building on this, Section III presents our data set. The results and implications for the ADRIAN research project are discussed in Section IV before we conclude our work.

II. Related Work

In the following, we review relevant work on profile interlinking (cf. Section II-A) and DTs (cf. Section II-B).

A. Profile Interlinking

The fundamental problem of profile interlinking is not new. The problem of identity matching was first mentioned by Newcombe in the late 1950s [11], long before the emergence of Web 2.0 and Internet use by the general public. The mathematical foundations for this followed ten years later by Fellegi and Sunter [12]. Since then, research has addressed this topic in the areas of databases, statistics, natural language processing, and data mining [13], among others.

Users systematically adapt their profile to the platform-specific standards with regard to language and wording in the profile. In doing so, they distinguish between formal and informal platforms, and even age- and gender-specific differences can be observed [2]. Even posts from the same person on different OSNs have linguistic variations. This is because users adapt their language style to the platform-specific norms [2]. Not only in terms of language, but also in terms of behavior, differences can be observed among users.

Recently, deep learning have also been applied to enable profile interlinking. Xu et al. [10] proposed an anchor node embedding method based on dual domain adaptation to learn the anchor node representation considering the attributes, topological structure and difference between domains. Users in different OSNs are called anchor nodes, and edges between users are called anchor links. In addition, they developed a node adaptation method based on a domain adaptation by backpropagation to learn the appropriate adaptation function using a backpropagation neural network. Moreover, Wang et al. [14] introduced a system called Fusion Embedding for User Identification (FEUI), in which user-pair graphs were interactively integrated by network structure, node attribute information, and node label. Thereby, the FEUI framework exploited a single-input and dual-output deep neural network to represent complex correlation from different information sources. Furthermore, Guo et al. [15] set up a deep neural tensor network-based model to represent the interactions between entities and extract the relationships between users from a higher dimension.

B. Digital Twin

The term DT is used in several areas of research and in practice. Among others, it appears in mechanical engineering, medicine, and computer science [16]. In artificial intelligence, the term has gained broader usage. In general, “DTs can be defined as (physical and/or virtual) machines or computer-based models that are simulating, emulating, mirroring, or ‘twinning’ the life of a physical entity, which may be an object, a process, a human, or a human-related feature” [16]. Here, we refer to the term as the digital representation of a human being that is created by personal information available on the Web [5]. The DT can never reflect the whole complexity of a real person, but represents characteristics that, alone or in combination with other characteristics, may pose a risk to the real person. Modeling DTs is based on established and freely available standards of the semantic web, such as Schema.org and FOAF (Friend of a Friend). At the same time, the overwhelming number of possible sources of information, the quality of the data, and a multitude of contradictory data make modeling challenging. However, studies [17] show that a large amount of relevant information is knowingly and unknowingly disclosed by users themselves [18].

III. Link Record & Profile Data

For the analysis of user behavior when posting cross-platform links (cf. Section III-A), data from YouTube was obtained as a starting point. In addition, Twitter data was collected based on the links included in YouTube videos to facilitate comparison. This allows us to determine which OSNs are interlinked and which OSNs are suitable entry points for data collection to create DTs. However, the relevance of the OSNs also comes from the trackable information. Again, what information can be found on multiple OSNs is crucial to ensure data quality through data matching. We therefore compare the different data points provided by the OSNs in Section III-B.

A. Link Record Analysis

The first goal is to analyze the collected data from YouTube and Twitter (cf. Table I).

TABLE I
Descriptive statistics: YouTube and Twitter dataset

YouTube	#	Twitter	#
Total Videos	4,605	Total Tweets	345,748
Total Channels	2,841	Total Users	842
Total Links	32,464	Total Links	467,834
Total Videos with Link	4,108	Total Tweets with Link	345,748
Videos/Channel (min)	1	Tweets/User (min)	1
Videos/Channel (mean)	1.62	Tweets/User (mean)	411.12
Videos/Channel (max)	39	Tweets/User (max)	87,343
Links/Video (min)	0	Links/Tweet (min)	1
Links/Video (mean)	6.94	Links/Tweet (mean)	1.35
Links/Video (max)	88	Links/Tweet (max)	8
Links/Channel (min)	1	Links/User (min)	1
Links/Channel (mean)	11.43	Links/User (mean)	607.58
Links/Channel (max)	513	Links/User (max)	174,356

We have collected a total of 4,605 videos belonging to 2,841 channels. On average, a video contains 6.94 links, a tweet has significantly fewer links than that, with 1.35 on average. The Twitter dataset has significantly more links with 607.58 links per user compared to 11.43 links per channel. This is obviously due to the fact that we have more tweets per user than videos per channel. Building on this, we turn to a deeper analysis of the links to better understand the link structure. We focus here on YouTube’s video descriptions, which allow users to insert a large number of links.

TABLE II
Number of links referring to a specific domain

Domain in YouTube Videos	# of Links	Domain in Tweets	# of Links
YouTube	4,647	Twitter	277,058
Bit	4,285	Screammov	87,054
Instagram	4,258	Trib	20,772
Twitter	2,334	Independent	11,061
Amazon	1,441	Bit	7,634
Facebook	1,415	TheGuardian	5,904
TikTok	903	LiverpoolEcho	4,881
Twitch	826	WioNews	4,567
Discord	540	FoxNews	2,983
Lnk	447	YouTube	2,925

Table II shows the domain distribution for links from YouTube videos and tweets. Most of the links in the YouTube videos lead to Instagram, Twitter, and Facebook. This correlates with the social media platforms relevant to our use case. For this reason, we analyze these three OSNs in more detail. In this context, it is important to determine how many links to the various platforms are contained in all videos or per channel.

TABLE III
ANALYSIS OF THE LINKS INCLUDED IN YOUTUBE VIDEOS

# of Links per Video/Channel	**on Twitter**	**on Facebook**	**on Instagram**
= 1	1,044	785	1,611
> 1	261	76	436
> 2	37	13	237
> 3	28	6	150
> 4	24	4	106
> 5	17	2	71
> 6	14	2	51
> 7	6	1	38
> 8	5	1	28
> 9	3	1	22
> 10	2	1	19

As shown in Table III, Instagram has the most links within embedded videos from YouTube in the descriptions, followed by Twitter and Facebook. Also, for any number between 2 and 10 links, Instagram achieves the highest number. It is also important to determine how many links overlap. A YouTube channel that contains links to all three OSNs (cf. Figure 1) is particularly relevant for creating a cross-platform profile.

B. *Profile Data Overview*

We divide the data into the following categories: Channel/User Identity, Channel/User Information, Content Information, Links to Images or Videos, External Links, Location Information, and Channel/User Metrics. The categories represent different aspects of user profiles, which generally require separate methods for correlation. The first category is used to establish the identity of a person. For this purpose, the name and username provided by the OSNs are used. The channel title or username appears in all OSNs, while the real name only appears on Instagram and Twitter. The challenge here is to determine, first, whether the user has provided the real name, and second, whether the username includes, for example, the user's first or last name. The next category is mainly about textual information for the channel/user. From the description of a profile in connection with the identity, it can be derived whether it is a person or an organization. In the case of persons, for example, the profession, interests, or further external links are revealed. The next category represents the content published by the channel or user. Tweets and Facebook posts are textual content enriched with images or videos, while YouTube and Instagram are images or videos with a description. The tags can provide initial information about the content and then form the basis for analyses over a period of time that provide insights into the behavior of the channel/user. The next two categories can provide very strong evidence for correlation. Images are suitable for a direct comparison of user profiles in different OSNs. The same applies to external links, e.g., when the profiles link to each other. Location data is available in all OSNs. Usually, except for Twitter, only the country or city is given as text. In the case of Twitter, many other types of representation are available, such as latitude and longitude coordinates, a bounding box, or even automatic extraction of locations from published content. Finally, the reach of a channel/user is also important. The number of followers or following can be used to determine user activity. In the case of Twitter, followers can be retrieved directly and direct connections between users can be analyzed.

TABLE IV
PROFILE DATA OVERVIEW FOR DIFFERENT OSNS

YouTube	**Instagram**	**Facebook**	**Twitter**
user_id	user_id	user_id	user_id
title	username	username	username
✗	fullName	✗	name
description	biography	about	description
publishedAt	✗	✗	created_at
privacyStatus	private	✗	✗
✗	verified	✗	verified
topicCategories	✗	category	✗
✗	✗	type	✗
post_id	post_id	post_id	post_id
publishTime	timestamp	timestamp	created_at
title	caption	✗	✗
description	✗	text	text
defaultLanguage	✗	✗	lang
madeForKids	✗	✗	possibly_sensitive
tags	hashtags	✗	hashtags
viewCount	✗	✗	retweet_count
commentCount	commentsCount	✗	reply_count
likeCount	likesCount	likes	like_count
kind	type	✗	✗
url	images	images	✗
externalUrl	displayUrl	✗	profile_image_url
✗	profilePicUrl	✗	media
✗	externalUrl	link	url
✗	✗	✗	entities.urls
✗	facebookPage	✗	✗
✗	✗	✗	location
✗	✗	✗	annotations
✗	✗	✗	coordinates
country	✗	✗	country_code
✗	✗	✗	place_type
✗	locationName	places_lived	full_name
subscriberCount	followersCount	followers	followers_count
✗	followsCount	following	following_count
videoCount	postCount	✗	tweet_count
viewCount	✗	✗	✗

IV. DISCUSSION

With limited resources and time, it is not possible for attackers to monitor OSNs live or to compare all existing profiles and keep the findings up to date. For this reason, in the ADRIAN project, we rely on OSN users to leave digital traces that point us to additional profiles and information that can be used to create DTs. However, this is not a disadvantage, but exactly the goal of the project. The point is not to create DTs on a daily basis, but to show that traces on the Web make this possible to some extent and enable threats like doxing. In this short paper, we were interested in finding out what traces can be found based on links and whether there are actually enough links to jump from one user profile to another to gather information and the intersections are significant.

Out of 2,841 profiles we identified using YouTube videos, we were able to infer three other OSNs in 507 cases using the links in the video descriptions (cf. Figure 1).

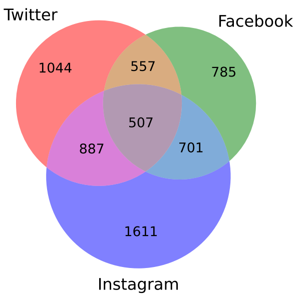


Fig. 1. Profile distribution across different OSNs

In 2,145 cases, we were able to find links to at least one other profile. The link behaviour of users is already a first data point used for identification. Beyond that, however, the large amount of different information about the respective OSNs is also worth mentioning (cf. Table IV). Not all information is available on all OSNs and in the same format, but it is often possible to merge them. In particular, user names, locations, geospatial information, and names or parts of names are often very good clues that help to interlink profiles.

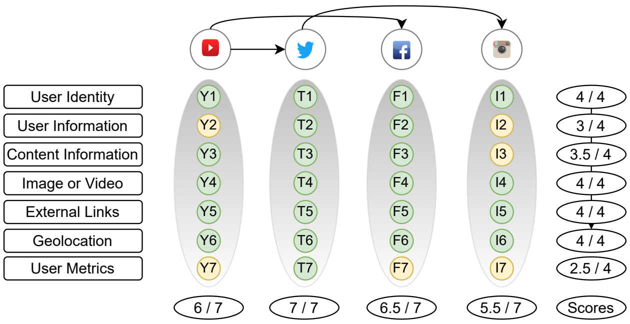


Fig. 2. Different profiles of a user in OSNs

We would like to discuss this with a real-world example – a random YouTube user who maintains multiple links in his video descriptions (cf. Figure 2): The YouTube channel contains a link to Facebook and Twitter profiles. For example, on Facebook, we can determine whether the profile belongs to an individual or an organization. This is important because the creation of the DT initially focuses on individuals. Moreover, the user is active on Twitter and posts, for example, sports activities that lead directly to Strava, an OSN for tracking physical activities that also includes social networking features. Since the source of his tweets is Instagram, the user also indicates another social media account that he uses. On Instagram, he posts photos that contain a lot of private information. As this example shows, the consolidation of the different profiles in OSNs is of considerable importance for the creation of a DT. On the one hand, it enables the validation of information and, on the other hand, information gaps can be closed. As the number of profiles in different OSNs increases, it can be assumed that a higher overall quality of the DT can be achieved.

V. Conclusion

We highlighted here that it is possible to infer from user profiles to other profiles. We showed that YouTube is particularly well suited as an entry portal for data acquisition, since it is possible to include many links to other OSNs in descriptions and this is also done regularly by users. In future work, we will match information from different profiles to initialize DTs.

Acknowledgment

This research is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr.

References

[1] P. Zhang, H. Zhu, T. Lu, H. Gu, W. Huang, and N. Gu, "Understanding relationship overlapping on social network sites: A case study of weibo and douban," *Proc. ACM Hum. Comput. Interact.*, vol. 1, no. CSCW, pp. 120:1–120:18, 2017.

[2] C. Zhong, H. Chang, D. Karamshuk, D. Lee, and N. Sastry, "Wearing many (social) hats: How different are your different social network personae?" in *Proc. of the 11th Intl. Conf. on Web and Social Media, ICWSM 2017*. AAAI Press, 2017, pp. 397–406.

[3] Data Portal, January 2022. [Online]. Available: https://datareportal.com/reports/digital-2022-global-overview-report (Accessed 2022-04-01).

[4] M. Pelletier, A. Krallman, F. Adams, and T. Hancock, "One size doesn't fit all: a uses and gratifications analysis of social media platforms," *JRIM*, vol. 14, no. 2, pp. 269–284, 2020.

[5] F. S. Bäumer, S. Denisov, Y. Su Lee, and M. Geierhos, "Towards Authority-Dependent Risk Identification and Analysis in Online Networks," in *Proc. of the IST-190 Research Symposium (RSY) on AI, ML and BD for Hybrid Military Operations (AI4HMO)*, A. Halimi and E. Ayday, Eds., October 2021.

[6] A. Petit, S. Ben Mokhtar, L. Brunie, and H. Kosch, "Towards Efficient and Accurate Privacy Preserving Web Search," in *Proc. of the 9th Workshop on Middleware for Next Generation Internet Computing*, 2014, pp. 1–6.

[7] M. Fire, R. Goldschmidt, and Y. Elovici, "Online Social Networks: Threats and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2019–2036, 2014.

[8] M. Chen, A. S. Y. Cheung, and K. L. Chan, "Doxing: What adolescents look for and their intentions," *IJERPH*, vol. 16, no. 2, p. 218, 2019.

[9] Y. Karimi, A. Squicciarini, and S. Wilson, "Automated detection of doxing on twitter," *arXiv preprint arXiv:2202.00879*, 2022.

[10] B. Xu, Y. Kou, G. Wang, D. Shen, and T. Nie, "Duallink: Dual domain adaptation for user identity linkage across social networks," in *Web Information Systems and Applications*, C. Xing, X. Fu, Y. Zhang, G. Zhang, and C. Borjigin, Eds. Cham: Springer, 2021, pp. 16–27.

[11] H. B. Newcombe, J. M. Kennedy, S. J. Axford, and A. P. James, "Automatic linkage of vital records," *Science*, vol. 130, no. 3381, pp. 954–959, 1959.

[12] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *J. Am. Stat. Assoc.*, vol. 64, no. 328, p. 1183, 1969.

[13] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 1, pp. 1–16, 2007.

[14] L. Wang, Y. Zhang, and K. Hu, "FEUI: Fusion Embedding for User Identification across social networks," *APIN*, pp. 1–17, 2021.

[15] X. Guo, Y. Liu, X. Meng, and L. Liu, "User Identity Linkage Across Social Networks Based on Neural Tensor Network," in *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 01 2021, pp. 162–171.

[16] B. R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," *IEEE Access*, vol. 7, pp. 167 653–167 671, 2019.

[17] F. S. Bäumer, J. Kersting, M. Orlikowski, and M. Geierhos, "Towards a multi-stage approach to detect privacy breaches in physician reviews." in *SEMANTICS Posters&Demos*, 2018.

[18] F. S. Bäumer, N. Grote, J. Kersting, and M. Geierhos, "Privacy matters: detecting nocuous patient data exposure in online physician reviews," in *ICIST*. Springer, 2017, pp. 77–89.

Patterns for Quantum Error Handling

Martin Beisel, Johanna Barzen, Frank Leymann, Felix Truger, Benjamin Weder, Vladimir Yussupov
Institute of Architecture of Application Systems, University of Stuttgart
Universitätsstrasse 38, 70569 Stuttgart, Germany
{firstname.lastname}@iaas.uni-stuttgart.de

***Abstract*—The capabilities of current quantum computers are limited by their high error rates. Thus, reducing the impact of these errors is one of the crucial challenges for the successful execution of quantum algorithms. For this purpose, various error handling methods have been proposed, ranging from error correction codes that detect and correct errors during the execution on a quantum device to post-processing techniques that mitigate errors classically. As these methods have different requirements and advantages, developers need to have a thorough understanding of them, to be able to select a suitable error handling method for their scenario. In this work, we present three new patterns for quantum error handling, describing proven solution strategies in a well-structured manner and integrate them into an existing quantum computing pattern language.**

Keywords-Quantum Computing; Pattern Language; Error Handling; Error Mitigation; Error Correction.

I. INTRODUCTION

Recent advances in the development of quantum devices resulted in the emergence of publicly available quantum computers [1]. Quantum computers are expected to solve certain problems, e.g., in chemistry [2] or combinatorial optimization [3], more efficiently than any classical computer. This is possible because quantum computers can exploit quantum mechanical effects, such as superposition and entanglement, to gain a computational advantage. However, the capabilities of the current generation of quantum computers are limited by a variety of factors, e.g., the low number of qubits and high error rates [1][4]. Due to the high number of errors caused by different error sources, e.g., error-prone gate and measurement operations, the execution result's accuracy is limited [5]. To deal with these errors, different error handling techniques have been proposed. Error correction codes, such as Shor's 9-qubit code [6], can be used to detect and correct occurring errors [7][8][9]. However, error correction requires a significant number of additional quantum resources. Thus, so-called error mitigation methods have been developed that require little to no additional quantum resources. These methods focus on the reduction of the negative impact caused by certain error types, e.g., Tensor Product Noise Model (TPNM) [10] and Fixed Identity Insertion Method (FIIM) [11] for measurement errors and gate errors, respectively. To incorporate any of these error handling methods into quantum applications, quantum software engineers need to understand their concepts, so they can select suitable ones for the case at hand.

A well-established approach for the description and structuring of proven solutions for reoccurring problems was presented by Alexander et al. [12] in the form of *patterns*. Each pattern describes a problem and its context and forces. Then, a proven solution for the problem is presented in an abstract manner, making the pattern applicable to different scenarios. Multiple patterns of the same domain can be combined in a *pattern language*. A pattern language for quantum computing has been introduced by Leymann [13]. Although it has been continuously expanded since its introduction [14][15][16], the quantum computing pattern language does not contain any patterns for the handling of quantum errors yet.

In this work, we extend the quantum computing pattern language by introducing three new patterns that describe well-established solutions for the handling of quantum errors. By documenting these proven solutions in an easy-to-understand and well-structured manner, we provide knowledge about quantum error handling to a broader audience. The ERROR CORRECTION pattern describes how to detect and correct quantum errors during the quantum computation. As this approach is not always feasible, two more patterns are introduced that focus on the mitigation of the impact of occurring errors. First, the READOUT ERROR MITIGATION pattern describes, how the impact of measurement errors can be reduced. Second, the GATE ERROR MITIGATION pattern presents proven solutions for the mitigation of gate errors that occur during the quantum circuit execution.

This paper is structured as follows: Section II introduces fundamental terms to establish a common vocabulary and describes the used pattern format. Then, Section III presents the error handling patterns in detail. In Section V, the related work is discussed, and Section VI concludes this work.

II. FUNDAMENTALS AND PATTERN STRUCTURE

In this section, we present our pattern format and establish the guidelines for the pattern authoring process. Further, we provide the fundamental terms related to quantum error handling that establish a common vocabulary.

A. Pattern Format & Authoring Method

The pattern format is derived from previous work on quantum computing patterns [14][15][16] and other best practices used by researchers [12][13][17][18][19][20]. A pattern is identified by its *name* and a mnemonic *icon*. First, the *problem* solved by the pattern is briefly described in form of a short question. Then a detailed description of the pattern's *context* and its *forces* is presented. The *solution* section describes a possible solution with a corresponding sketch. The *result* paragraph explains the context following the application of

ISBN: 978-1-61208-953-9

the solution and discusses possible consequences. Afterwards, one or multiple *examples* of the previously introduced solution are explained textually and visually. In the *related patterns* section, the relationship of the pattern to other patterns within the pattern language is described. Finally, the *known uses* section lists implementations of the pattern.

For the identification of the quantum error handling patterns, we analyzed state-of-the-art approaches in scientific literature. Recurring solution strategies were collected, analyzed, and ultimately compiled into the quantum error handling patterns. Due to the lack of currently available code fragments implementing the error handling patterns, identifying a rich collection of concrete solutions remains future work. These can then be integrated into a future quantum computing solution language, facilitating the patterns' application [21].

B. Fundamental Terms

Quantum Device: A quantum device is a gate-based quantum computer, e.g., IBM's, Google's, or IonQ's quantum computers. Quantum devices can execute *quantum circuits* that implement *quantum algorithms*. Typically, device access is provided via the cloud. Some providers offer free access to small devices [22], however, access to state-of-the-art devices is costly and can scale with the number of executed quantum circuits and their size [23]. Further, the limited number of available devices, in combination with the high demand for usage, can lead to queue times. Due to the fragility of coherent quantum states, quantum devices are error-prone. This is particularly true for currently available Noisy Intermediate-Scale Quantum (NISQ) devices.

Quantum Algorithm: A quantum algorithm is an algorithm that can be executed on a quantum device and typically makes use of quantum mechanical effects to achieve a computational advantage over its classical counterpart. The gate-based representation of a quantum algorithm is a quantum circuit. Most currently used quantum algorithms are hybrid quantum-classical algorithms that consist of a classical and a quantum part. Typical examples are Variational Quantum Algorithms (VQAs), such as Variational Quantum Eigensolver (VQE) [24] and the Quantum Approximate Optimization Algorithm (QAOA) [25]. VQAs employ shallow parameterized quantum circuits that are optimized classically, to perform meaningful computations on current NISQ devices [26].

Quantum Circuit: A quantum circuit is a model for quantum computation. Each step of the computation is modeled by a *quantum gate*. At the end of each circuit a *measurement* is performed, to retrieve the result of the quantum computation. Due to the probabilistic nature of quantum computing, the circuit has to be executed multiple times to obtain a reliable probability distribution. The depth of a circuit is the number of layers of 1- or 2-qubit gates in which parallel operations are performed on disjoint qubits. Its width is defined as the number of qubits involved in the computation.

Quantum State: A quantum state describes the current state of one or multiple qubits. While classical bits can be in the states 0 and 1, qubits can be in corresponding $|0\rangle$ and $|1\rangle$ states, however, they can also be in a superposition, i.e., a combination of both states. A quantum state can be measured to obtain a probability distribution for each possible state.

Ancilla Qubit: Ancilla qubits follow the concept of classical ancilla bits. They are additional qubits that are typically used temporarily to store information or achieve a specific goal, e.g., they can store entangled states.

Quantum Gate: Quantum gates are the elementary operations of a quantum circuit that can be executed on a quantum device's qubits. Quantum gates are unitary operations, which can be described mathematically by unitary matrices. Quantum devices only support a limited gate set, which is usually restricted to a small number of 1- or 2-qubit gates [27]. Furthermore, quantum devices can execute quantum gates only with limited accuracy, resulting in so-called gate errors [5]. As these errors can occur with every execution of a gate, they continuously accumulate during the quantum circuit execution.

Measurement: *Readout* or *observation* are common synonyms for measurement. The quantum state prepared by a quantum circuit is sampled by performing a measurement operation. By measuring a qubit, its state is collapsed and can not be restored. Therefore, a measurement operation is not a quantum gate. Moreover, measurement times are significant in comparison to gate times, causing delays that amplify the devices's decoherence [5]. Hence, measurements are one of the main error sources of a quantum device [28].

Quantum Error: Quantum errors, also known as *Noise*, are one of the key limitations in the current era of quantum computing. The capabilities of quantum devices are limited by the error-prone and highly fragile quantum states that are used for computation, as the occurrence of too many errors makes the measurement result unusable. First, the aforementioned quantum gate errors can occur with every execution of a gate. Thereby, the error rates of multi-qubit gates are particularly high [27]. These errors account for a significant part of the overall error and limit a quantum circuit's depth. Second, unintended bit-flips occurring during the measurement, lead to incorrect measurement results. To execute a quantum circuit on a quantum device, the circuit needs to be compiled for the device's gate set and qubit connectivity mapping [29]. As each of the qubits usually only has a direct connection to a small subset of the other qubits, a lot of SWAP operations may be required to establish multi-qubit operations foreseen in the uncompiled quantum circuit [4]. Such swap operations lead to additional 2-qubit gates and increase the circuit depth. Thus, designing a circuit for a specific device or vice versa can prevent errors. Additionally, qubits can unintentionally influence the state of other qubits [5]. This so-called crosstalk is difficult to predict and can further increase the overall error. Moreover, quantum states can decohere, meaning that after a device-dependent amount of time, they decay irreversibly due to environmental influences. Hence, all quantum gates and measurement operations have to be performed before the state decoheres, directly limiting a circuit's maximum depth.

III. Patterns for Quantum Error Handling

In this section, we introduce three new patterns focusing on error handling for quantum devices, namely Error Correction, Gate Error Mitigation, and Readout Error Mitigation. These patterns focus on the prevention and reduction of errors occurring during the quantum circuit execution and extend the existing quantum computing pattern language [13][14][16]. First, we present the error handling pattern category in the context of the quantum computing patterns and then introduce each new pattern in detail.

A. *Quantum Computing Patterns for Error Handling*

The quantum computing patterns form a pattern language supporting quantum software engineers in developing quantum applications. As most quantum applications are hybrid [4][30] the focus is on hybrid-classical algorithms. The quantum computing patterns capture reoccurring problems in the domain of quantum computing and provide proven solutions for them. Figure 1 shows the basic structure of a hybrid algorithm [4] and an overview of the quantum computing pattern language. The process starts off, by pre-processing data on a classical computer. A typical example is the preparation of the data required for state preparation. The state preparation routine prepares the quantum computer's initial state, e.g., a uniform superposition can be created, or the previously prepared data can be encoded into the initial state. This can be done by employing one of the quantum state or data encoding patterns. The prepared quantum state can then be manipulated by applying unitary transformations. These perform the quantum algorithm's operations. Typical operations are explained in detail in the unitary transformation patterns [13]. Best practices for realizing quantum algorithms are described in the program flow patterns [16]. The last step performed on the quantum computer is the measurement operation. Thereby, the quantum algorithm's final state is retrieved in the form of a probability distribution. The measurement result can then be post-processed, e.g., performing continued fraction expansion for Shor's algorithm [31]. In the case of a VQA, the hybrid algorithm has a loop. Thereby, the result is incorporated into the next iteration unless its termination condition is fulfilled. Thus, the quantum computing patterns cover the entire cycle of a hybrid quantum algorithm.

However, in their current state, the quantum computing patterns do not address the handling of errors. Since quantum errors are one of the major factors limiting quantum computing, particularly in the current NISQ era, reducing their negative impact is essential. Due to the variety of error sources and hardware limitations, different solution strategies have evolved. The Error Correction pattern focuses on the in-flight repair of computational errors. It enables fault-tolerant large-scale quantum computing by detecting and fixing errors immediately during the circuit execution. However, error correction requires a large number of quantum resources that are not available on current quantum devices yet. A NISQ-compatible alternative to error correction is error mitigation. Error mitigation tolerates the occurrence of errors and focuses on the reduction of their impact. In particular, the Readout Error Mitigation and Gate Error Mitigation pattern document two different kinds of error mitigation, focusing on the reduction of the impact of measurement and gate errors.

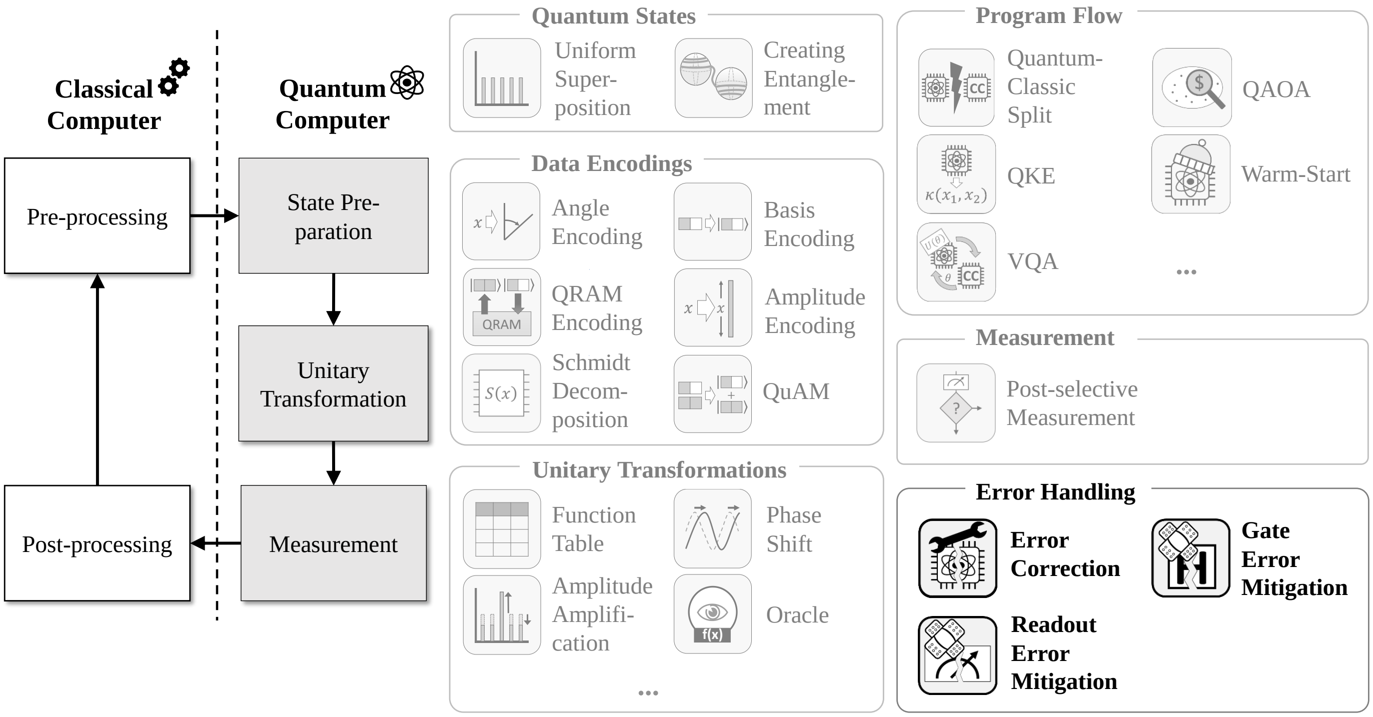


Figure 1. Overview of a hybrid algorithm's typical structure and existing and new (in bold) quantum computing patterns, extends [14][16].

B. Error Correction Pattern

How to detect and correct errors occurring during the execution of a quantum circuit?

Context: A quantum algorithm needs to be run on a quantum device. The quantum device's performance is limited by various error sources, such as gate errors and crosstalk. The prevention of these errors enables the execution of large-scale quantum algorithms for real-world problems.

Forces: Quantum devices unavoidably cause a certain amount of errors due to the fragility of coherent quantum states [7][32]. Furthermore, contrary to classical bits, qubits can not be copied [33]. Hence, classical error correction can not be used for quantum computers and new quantum-specific methods need to be developed. However, these methods can be costly in terms of quantum resources, as they require a large number of additional qubits and quantum gates.

To enable scalable quantum computing for real-world problems, all kinds of errors occurring in quantum devices need to be detected and corrected. In general, the correction of errors is preferred over their mitigation, since even minor remaining post-mitigation errors slowly stack up during the computation and ultimately lead to an imprecise result.

Solution: Detect and correct quantum errors using quantum error correction codes [7][34][35][36], which are added to the executed circuit. With these correction codes, many physical qubits are combined into one logical qubit. As a result of this bundling, errors in the original qubit can be first detected and then corrected [4]. Figure 2a depicts a solution sketch showcasing the general building blocks of a quantum error correction procedure. The shown instance applies an error correction code that can detect and fix bit-flip errors in the computational basis. For the correction of errors from other sources, similar processes can be applied. First, the *ancilla coupling* is created, by encoding the state $|\psi\rangle$ of a single physical qubit into multiple ancilla qubits. These qubits now hold the logical qubit's data and are called *data qubits* in the following. Next, some unitary transformation is applied to the logical qubit, possibly resulting in an error. In order to *detect* an error, additional ancilla qubits are employed to check the parity of the data qubits. Based on the discovered syndrome, the error-free state can be recovered in the *recovery phase*. Note that the process is assumed to only have errors at the unitary transformation step, which is denoted by the error indicator. Further, the number and type of detectable errors depends on the applied error correction code.

Result: When applying quantum error correction, computational errors can be prevented, enabling error-free systems of logical qubits. Thus, error correction is making fault-tolerant quantum computation feasible. The good scalability of error correction, enables the accurate execution of large algorithms.

Examples: Figure 2b illustrates the application of a 3-qubit variant of the aforementioned error code for multiple qubits. Each of the physical qubits P1 to P4 is transformed into a logical qubit consisting of five physical qubits. Three of these five physical qubits are being used as data qubits and two of them are being used for the detection and recovery process. Further, the 1- and 2-qubit gates G1 to G4 need to be realized by the subroutines S1 to S4, which prepare the data qubits. The resulting errors can then be corrected by individually applying error correction routines for each of the logical qubits.

Related Patterns: Instead of preventing quantum errors, the READOUT ERROR MITIGATION and GATE ERROR MITIGATION patterns focus on reducing the errors' negative impact on results. This pattern can be applied to quantum algorithms, e.g., QAOA pattern.

Known Uses: Laflamme et al. [8] show a 5-qubit error correction code that can protect a qubit against general 1-qubit errors. Shor's 9-qubit code can protect a qubit against single bit-flip and phase-flip errors [6]. Further, a variety of different quantum error correction codes have been presented in the literature [7][34][35][36].

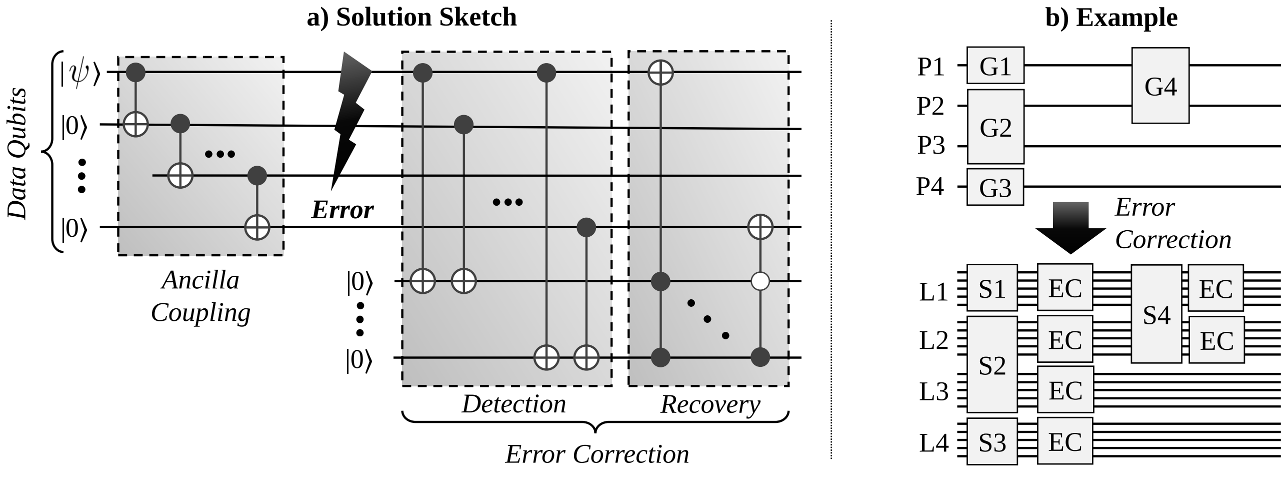


Figure 2. Solution sketch for the ERROR CORRECTION pattern and example of error correction with a 3-qubit bit-flip code for a 4-qubit circuit [4][7].

C. Readout Error Mitigation Pattern

How to reduce the impact of erroneous measurements such that the measured result is closer to the intended quantum state?

Context: A NISQ-compatible quantum algorithm, e.g., QAOA or VQE, needs to be run on a quantum device. The device's decoherence times are short and the measurement operations are error-prone. Hence, the measured probability distribution is inaccurate, even when the measured quantum state is accurate. Thus, the negative impact of readout errors needs to be mitigated to obtain a precise measurement result.

Forces: The measurement times of quantum computers in the NISQ era are significant in comparison to their decoherence times [37]. Therefore, the measurements are highly error-prone and often are among the main error sources [10]. Due to the limited capabilities of current NISQ devices, a minimal number of additional qubits and quantum gates shall be used for the mitigation of readout errors. Further, a quantum device's measurement error rates change over time, thus the Readout Error Mitigation (REM) needs to be adaptive.

Solution: Mitigate the impact of readout errors by applying a REM method. The mitigation method is performed after the circuit execution and adjusts the measured probability distribution. The resulting mitigated probability distribution is a more accurate representation of the intended quantum state. A solution sketch for the application of REM is shown in Figure 3a. First, the quantum circuit is implemented and executed. Then the resulting probability distribution is improved based on measurement characteristics collected for the quantum device. These characteristics are typically obtained by separately running so-called calibration circuits. Alternatively, adapted instances of the implemented circuit can be run to obtain additional information about the measurement properties.

Result: REM can reduce the impact of errors caused by measurement operations. The resulting, more precise probability distributions make NISQ devices more suitable for real-world use cases. However, additional classical processing is necessary, which can significantly increase the runtime and classical resource requirements, as not all mitigation methods scale well with the number of qubits. Generally, data provenance can be employed to increase the efficiency of frequently occurring REM tasks, e.g., when executing a VQA.

Examples: Figure 3b illustrates the steps of the Static Invert-and-Measure (SIM) [28] technique. First, multiple slightly adapted instances of the circuit are created. Thereby, bit-flips are added right before the circuit's measurement operations. This helps to detect erroneous measurements because readout error rates are typically higher when measuring a qubit in the $|1\rangle$ state than when measuring it in the $|0\rangle$ state [28]. Once all circuits are executed, the measurement results are processed, returning the mitigated probability distribution. Figure 3c shows the typical process of a calibration matrix-based mitigation method. Multiple shallow calibration circuits are generated and executed. The resulting probability distributions give information about the device's readout error rates. These error rates are then incorporated into a so-called calibration matrix, which can be used to mitigate readout errors. For example, this can be done by multiplying the inverse of the calibration matrix with the circuit's measurement result.

Related Patterns: The GATE ERROR MITIGATION pattern can be used in combination with this pattern for more extensive mitigation. This pattern can be applied to hybrid quantum algorithms, e.g., VQE or QAOA pattern. This pattern commonly uses BASIS ENCODING pattern to generate calibration circuits.

Known Uses: Various REM methods, e.g., calibration matrix-based [10][37][38][39] or bit-flip-based [28][40][41], have been introduced in the literature. Moreover, recent work introduces a deep learning-based REM method [42].

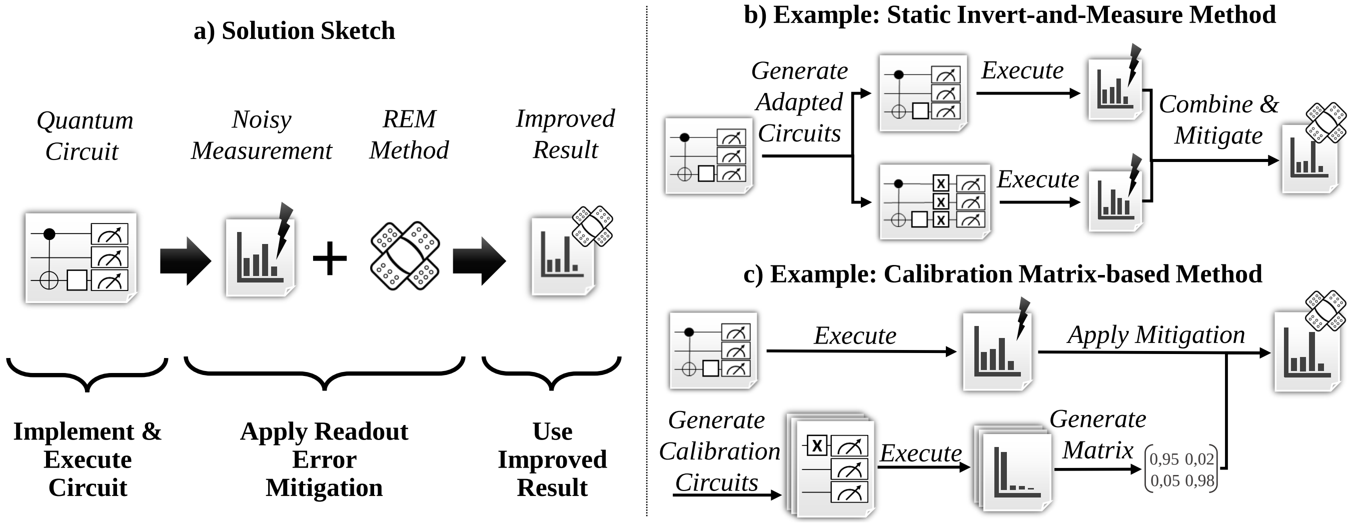

a) Solution Sketch
Quantum Circuit
Noisy Measurement
REM Method
Improved Result
+
Implement & Execute Circuit
Apply Readout Error Mitigation
Use Improved Result
b) Example: Static Invert-and-Measure Method
Generate Adapted Circuits
Execute
Execute
Combine & Mitigate
c) Example: Calibration Matrix-based Method
Execute
Apply Mitigation
Generate Calibration Circuits
Execute
Generate Matrix
0,95 0,02
0,05 0,98

Figure 3. Solution sketch for the READOUT ERROR MITIGATION pattern and two example processes for bit-flip- and calibration matrix-based methods.

D. Gate Error Mitigation Pattern

How to reduce the negative impact of noisy gate executions such that the pre-measurement state is closer to the expected error-free state?

Context: A NISQ-compatible quantum algorithm, e.g., VQE, needs to be run on a quantum device. The device's gate implementations are error-prone, causing errors in the quantum computation. To obtain precise results for the executed algorithm, the measured state needs to be computed accurately. Thus, it is crucial to mitigate the effects of gate errors.

Forces: The execution of gates on current NISQ devices is not perfectly accurate. Hence, every execution of a gate causes a minor error. These errors keep accumulating, eventually making large computations impossible. The pulses used for the implementation of gate operations can be controlled on many quantum devices [43][44]. Therefore, custom pulse schedules can be used to individually calibrate gates. Furthermore, the capabilities of current quantum devices are limited, e.g., the number of qubits and the decoherence times are bound. Thus, minimal additional quantum resources, such as gates and qubits, shall be used for error mitigation.

Solution: Mitigate the impact of gate errors by applying a Gate Error Mitigation (GEM) method. The mitigation of gate errors has to be performed before the execution of the quantum circuit, as occurring errors otherwise accumulate during the computation, making it difficult to retrace them. The resulting pre-measurement quantum state is closer to the expected error-free state, therefore, providing more accurate measurement results. Figure 4a depicts a solution sketch for GEM. First, the circuit is implemented. Afterwards, a GEM method is applied, modifying the circuit, to generate a more precise implementation for the selected device. The circuit modifications can range from simple gate additions over custom gate pulse adjustments to full circuit rewrites based on Machine Learning (ML). Next, the improved circuit is executed on the quantum device. Finally, the improved measurement result can be evaluated to obtain a more precise solution.

Result: GEM can significantly reduce the impact of errors caused by erroneous gate executions. As a consequence, the state computed by the quantum algorithm is closer to the expected error-free quantum state and a more precise algorithm result can be obtained. However, the mitigation process may induce additional quantum gates into the circuit or require classical pre-processing to calculate optimal device calibrations, e.g., gate pulse calibrations. Generally, GEM methods can be used in combination with other error mitigation methods, such as REM to reduce the overall error further.

Examples: Figure 4b shows the process of a typical gate addition-based method. These methods mitigate gate errors by adding additional gates to the quantum circuit that balance out gate errors. The initial quantum circuit is modified by adding specific gates for each error-prone operation. Hence, the depth of the circuit increases significantly. Therefore, the device's decoherence times need to be kept in mind, as otherwise, the mitigation might decrease the result quality.

Figure 4c depicts the typical process of a pulse calibration method. First, the pulse calibrations for the device are generated, e.g., it is determined which frequency is perfect to perform a bit-flip operation on a specific qubit. Once all required frequencies are determined, the information can be incorporated into the quantum circuit. When executing the modified circuit, the custom pulse calibrations will now be used instead of the default values. More precise pulse calibrations make gate executions more accurate, thus, decreasing gate error rates and increasing the solution's precision.

Related Patterns: This pattern can be used in combination with the READOUT ERROR MITIGATION pattern to further reduce the overall error. This pattern can be applied to hybrid quantum algorithms, e.g., VQE or QAOA pattern.

Known Uses: Several circuit adjustment methods, e.g., FIIM or Random Identity Insertion (RIIM), are presented in the literature [11][45][46]. Further, machine learning-based circuit adjustment methods have been introduced, e.g., Noise-Aware Circuit Learning (NACL) [47]. Moreover, pulse modification-based GEM methods have been presented [48][49].

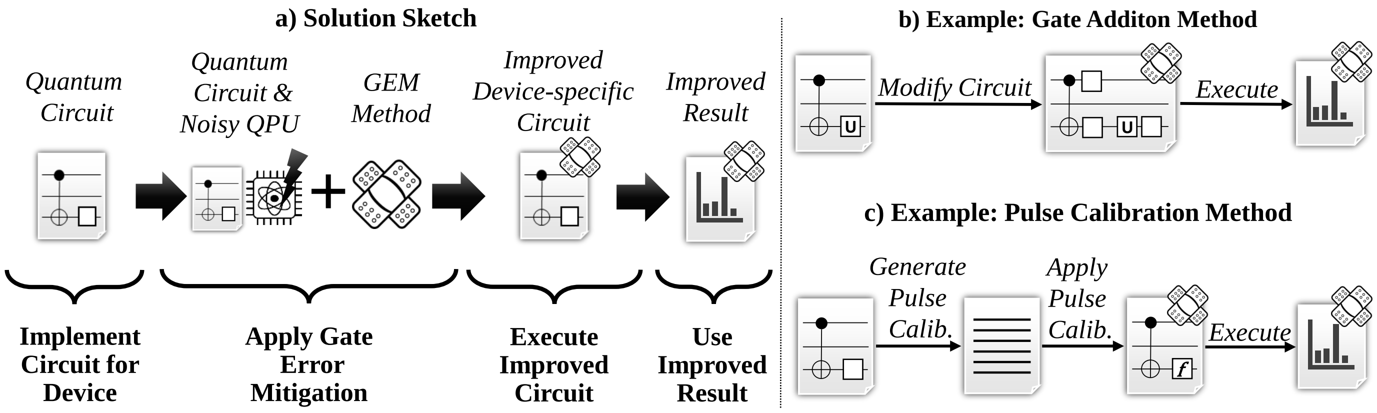

a) Solution Sketch
Quantum Circuit
Quantum Circuit & Noisy QPU
GEM Method
Improved Device-specific Circuit
Improved Result
U
Implement Circuit for Device
Apply Gate Error Mitigation
Execute Improved Circuit
Use Improved Result
b) Example: Gate Additon Method
Modify Circuit
Execute
c) Example: Pulse Calibration Method
Generate Pulse Calib.
Apply Pulse Calib.
Execute
f

Figure 4. Solution sketch for the GATE ERROR MITIGATION pattern and two example processes for a gate addition and a pulse calibration method.

IV. Pattern Validation and Discussion

In software engineering, a pattern's validity can be confirmed by showing the existence of a large enough number of real-world occurrences [17][19]. For each presented quantum error handling pattern there exist multiple distinct real-world occurrences as shown in Table I.

The Error Correction pattern is implemented on a basic level by Laflamme's 5-qubit code [8] and Shor's 9-qubit code [6] which protect a single qubit against general 1-qubit errors, and bit- and phase-flips, respectively. Consequently, more advanced methods emerged that rely on such basic error correction implementations, e.g., subsystem or topological error correction codes [7]. Some examples include Bacon-Shor (BS) [50], Calderbank-Shor-Steane (CSS) [35][51][52] and surface codes [36].

The Readout Error Mitigation pattern's most used implementations rely on calibration matrices. Prominent examples are TPNM [10], Continuous Time Markov Processes (CTMP) [10], Matrix-free Measurement Mitigation (M3) [39], and Diagonal Detector Overlapping Tomography (DDOT) [53]. In contrast, the SIM [28], Adaptive Invert-and-Measure (AIM) [28], and Bit-Flip Averaging (BFA) [40] method retrieve measurement error rates by running additional circuits containing bit-flips.

The Gate Error Mitigation pattern can be implemented in the form of a gate addition method, such as Zero Noise Extrapolation (ZNE) [54]. Concrete examples of ZNE are FIIM [11], RIIM [11], List Identity Insertion Method (LIIM) [54], or Set Identity Insertion Method (SIIM) [54]. Other methods, e.g., NACL [47], reduce gate error rates based on ML-based circuit learning.

Table I. Real-world occurrences of the introduced patterns

Error Handling Pattern	Real-world Occurrences
Error Correction	Laflamme's 5-qubit code, Shor's 9-qubit code, BS codes, Surface codes, CSS codes
Readout Error Mitigation	TPNM, CTMP, M3, DDOT, SIM, AIM, BFA
Gate Error Mitigation	FIIM, RIIM, LIIM, SIIM, NACL

Software-based handling of quantum errors is of utmost importance due to the high error rates of quantum devices in the NISQ era. However, it is expected that error rates will continue to decrease with every new generation of quantum devices, which raises the question of whether error handling will stay of importance in the long term. Due to the fragility of quantum states, it seems unlikely that the occurrence of quantum errors can be avoided entirely. Hence, quantum error handling will remain important and it is rather a question about *which new kinds of error handling techniques* will appear in the future. Error correction is expected to be the most promising long-term solution as it provides the possibility of fault-tolerant quantum computing once the hardware capabilities are sufficient. However, solely focusing on error correction would disregard the state of the current and near-term generation of quantum devices, as their limited capabilities make the application of error correction infeasible and require applying error mitigation techniques.

Furthermore, it is likely that hardware providers will integrate automatic error handling systems in the future. For instance, IBM already integrates the M3 REM method into Qiskit Runtime [55], an environment for running hybrid quantum algorithms close to the quantum device. However, solely relying on such systems limits developers to the options offered by hardware providers, as it is impossible to modify or extend provider APIs, e.g., by optimizing a method's implementation or integrating a newly published error handling method. Therefore, the introduced catalog of quantum error handling patterns, which can also be extended with new patterns documenting other kinds of error handling techniques, is also helpful as a structured guide that facilitates the general understanding of the topic.

V. Related Work

The patterns for quantum error handling introduced in this work extend the existing quantum computing pattern language [13][14][15][16]. Besides these patterns, there are other works that summarize reoccurring concepts in the quantum computing domain [56][57][58]. However, they are not following the pattern concept first introduced by Alexander et al. [12] in the architecture domain. The pattern concept is not restricted to architecture though, it is also widely spread in information technology. Examples are the enterprise integration patterns [59] or the cloud computing patterns [18]. To the best of our knowledge, there have been no other patterns published in the domain of quantum computing.

To facilitate the application of abstract solutions, Falkenthal and Leymann [21] introduced the concept of solution languages. Solution languages provide concrete solutions for specific patterns, e.g., a usable circuit or an implementation of a pattern for a specific language [60]. The concrete solution artifacts are linked to their corresponding patterns and connected to other solutions according to the relations of the pattern language [61]. Thus, concrete circuits and implementations of different methods solving one of the error handling patterns can be provided in a corresponding solution language.

With the continuous increase of transistors and clock speeds in classical computing, reliability has emerged as a critical concern [62]. Hence, novel error correction codes for classical hardware keep getting proposed [63]. It has been shown that concepts from classical error correction can be employed for quantum hardware [64], therefore, the emergence of new classical methods may prove useful to quantum error correction.

In the domain of quantum error handling, there have been multiple works surveying existing methods and explaining the basic concepts of error correction and error mitigation [4][7][37]. However, none of them is guiding users in applying error handling solutions to a given problem at hand. For example, Devitt et al. [7] briefly describe the fundamentals of error correction and then present a variety of methods in detail. Since these detailed method descriptions require a quantum computing background, they are not easy and fast to understand for people that are non-experts and mainly want to get an overview of how they can deal with quantum errors. On the contrary, our work provides easy access to well-structured compact knowledge artifacts, which facilitates a fast understanding of the topic.

VI. Conclusion and Future Work

Although quantum computing advanced rapidly in the last few years, the current generation of quantum devices is still highly error-prone. To achieve meaningful results, quantum software engineers need to understand the limitations and how to minimize the impact of the occurring errors. In this work, we extend the quantum computing patterns, by introducing three new patterns for quantum error handling that shall support quantum software engineers in building and running quantum algorithms successfully on error-prone quantum devices. As quantum computing is an interdisciplinary domain, we first introduce the fundamental terms to establish a common set of vocabulary, explaining the required basic concepts. Then, we present the quantum error handling patterns, explaining and showcasing proven solutions strategies for the prevention and mitigation of different types of quantum errors.

For future work, we plan to incorporate the quantum error handling patterns into PlanQK [65], a platform for sharing knowledge about quantum computing. PlanQK uses the pattern repository Pattern Atlas [66] for the presentation of patterns and contains all currently published quantum computing patterns [67]. By including all quantum computing patterns in such an online platform, a constant re-evaluation and a continuous evolution of the pattern language can be achieved. This also includes an evaluation of the usability of the error handling patterns based on the feedback provided by the community. The evaluation results can then be used to refine the patterns and further improve them. Ensuring a constant re-evaluation is of great importance for the quantum computing pattern language, as the domain is still rapidly evolving, and we expect the emergence of more best practices that we plan to abstract into new patterns that further extend the quantum computing pattern language. Additionally, we plan to implement a quantum computing solution language that will facilitate the application of the patterns presented in this work. The solution language is also planned to be integrated into PlanQK, enabling users to add code snippets and link them to the corresponding pattern.

Acknowledgment

This work was partially funded by the project *SEQUOIA* funded by the Baden-Wuerttemberg Ministry of Economic Affairs, Labour and Tourism and by the BMWK project *PlanQK* (01MK20005N)

References

[1] J. Preskill, “Quantum computing in the nisq era and beyond,” *Quantum*, vol. 2, p. 79, 2018.

[2] Y. Cao *et al.*, “Quantum chemistry in the age of quantum computing,” *Chemical Reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[3] E. Zahedinejad and A. Zaribafiyan, “Combinatorial optimization on gate model quantum computers: A survey,” *arXiv preprint arXiv:1708.05294*, 2017.

[4] F. Leymann and J. Barzen, “The bitter truth about gate-based quantum algorithms in the NISQ era,” *Quantum Science and Technology*, vol. 5, no. 4, p. 044007, 2020.

[5] M. Salm, J. Barzen, F. Leymann, and B. Weder, “About a Criterion of Successfully Executing a Circuit in the NISQ Era: What $wd \ll 1/\epsilon_{\text{eff}}$ Really Means,” in *Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software*, ser. APEQS 2020. ACM, 2020, p. 10–13.

[6] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A*, vol. 52, pp. R2493–R2496, 1995.

[7] S. J. Devitt, W. J. Munro, and K. Nemoto, “Quantum error correction for beginners,” *Reports on Progress in Physics*, vol. 76, no. 7, p. 076001, 2013.

[8] R. Laflamme, C. Miquel, J. P. Paz, and W. H. Zurek, “Perfect quantum error correcting code,” *Phys. Rev. Lett.*, vol. 77, pp. 198–201, 1996.

[9] R. Matsumoto and M. Hagiwara, “A survey of quantum error correction,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 104, no. 12, pp. 1654–1664, 2021.

[10] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, “Mitigating measurement errors in multiqubit experiments,” *Physical Review A*, vol. 103, no. 4, p. 042605, 2021.

[11] A. He, B. Nachman, W. A. de Jong, and C. W. Bauer, “Zero-noise extrapolation for quantum-gate error mitigation with identity insertions,” *Physical Review A*, vol. 102, no. 1, p. 012426, 2020.

[12] C. Alexander, *A pattern language: towns, buildings, construction*. Oxford university press, 1977.

[13] F. Leymann, “Towards a pattern language for quantum algorithms,” in *International Workshop on Quantum Technology and Optimization Problems*. Springer, 2019, pp. 218–230.

[14] M. Weigold, J. Barzen, F. Leymann, and M. Salm, “Encoding patterns for quantum algorithms,” *IET Quantum Communication*, vol. 2, no. 4, pp. 141–152, 2021.

[15] ——, “Expanding data encoding patterns for quantum algorithms,” in *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, 2021, pp. 95–101.

[16] M. Weigold, J. Barzen, F. Leymann, and D. Vietz, “Patterns for hybrid quantum algorithms,” in *Symposium and Summer School on Service-Oriented Computing*. Springer, 2021, pp. 34–51.

[17] J. O. Coplien and A. W. O. Alexander, “Software patterns,” 1996.

[18] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud computing patterns: fundamentals to design, build, and manage cloud applications*. Springer, 2014.

[19] E. Gamma, R. Helm, R. Johnson, J. Vlissides, and D. Patterns, *Elements of reusable object-oriented software*. Addison-Wesley Reading, Massachusetts, 1995, vol. 99.

[20] V. Yussupov, J. Soldani, U. Breitenbücher, A. Brogi, and F. Leymann, “From serverful to serverless: A spectrum of patterns for hosting application components.” in *CLOSER*, 2021, pp. 268–279.

[21] M. Falkenthal and F. Leymann, “Easing pattern application by means of solution languages,” in *Proceedings of the 9th International Conference on Pervasive Patterns and Applications*, 2017, pp. 58–64.

[22] IBM, “Overview of ibm quantum computing systems,” https://www.ibm.com/quantum-computing/systems/ [accessed: 2022.01.24].

[23] Amazon, “Overview of Amazon Braket Pricing,” https://aws.amazon.com/de/braket/pricing/ [accessed: 2022.01.24].

[24] A. Peruzzo *et al.*, “A variational eigenvalue solver on a photonic quantum processor,” *Nature communications*, vol. 5, no. 1, pp. 1–7, 2014.

[25] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” *arXiv preprint arXiv:1411.4028*, 2014.

[26] M. Cerezo *et al.*, “Variational quantum algorithms,” *Nature Reviews Physics*, pp. 1–20, 2021.

[27] S. Sivarajah *et al.*, “t| ket⟩: a retargetable compiler for NISQ devices,” *Quantum Science and Technology*, vol. 6, no. 1, p. 014003, 2020.

[28] S. S. Tannu and M. K. Qureshi, “Mitigating measurement errors in quantum computers by exploiting state-dependent bias,” in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2019, p. 279–290.

[29] M. Salm *et al.*, “The NISQ Analyzer: Automating the selection of quantum computers for quantum algorithms,” in *Symposium and Summer School on Service-Oriented Computing*. Springer, 2020, pp. 66–85.

[30] B. Weder, J. Barzen, F. Leymann, and D. Vietz, “Quantum software development lifecycle,” in *Quantum Software Engineering, arXiv preprint arXiv:2106.05800*. Springer, 2022.

[31] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM review*, vol. 41, no. 2, pp. 303–332, 1999.

[32] J. Chiaverini *et al.*, “Realization of quantum error correction,” *Nature*, vol. 432, no. 7017, pp. 602–605, 2004.

[33] W. K. Wootters and W. H. Zurek, “The no-cloning theorem,” *Physics Today*, vol. 62, no. 2, pp. 76–77, 2009.

[34] W. Cai, Y. Ma, W. Wang, C.-L. Zou, and L. Sun, “Bosonic quantum error correction codes in superconducting quantum circuits,” *Fundamental Research*, vol. 1, no. 1, pp. 50–67, 2021.

[35] D. Gottesman, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

[36] J. Roffe, “Quantum error correction: an introductory guide,” *Contemporary Physics*, vol. 60, no. 3, pp. 226–245, 2019.

[37] B. Nachman, M. Urbanek, W. A. de Jong, and C. W. Bauer, “Unfolding quantum computer readout noise,” *npj Quantum Information*, vol. 6, no. 1, pp. 1–7, 2020.

[38] F. B. Maciejewski, Z. Zimborás, and M. Oszmaniec, “Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography,” *Quantum*, vol. 4, p. 257, 2020.

[39] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, “Scalable mitigation of measurement errors on quantum computers,” *PRX Quantum*, vol. 2, no. 4, p. 040326, 2021.

[40] A. W. Smith, K. E. Khosla, C. N. Self, and M. Kim, “Qubit readout error mitigation with bit-flip averaging,” *arXiv preprint arXiv:2106.05800*, vol. 7, no. 47, p. eabi8009, 2021.

[41] M. Streif, M. Leib, F. Wudarski, E. Rieffel, and Z. Wang, “Quantum algorithms with local particle-number conservation: Noise effects and error correction,” *Physical Review A*, vol. 103, no. 4, p. 042412, 2021.

[42] S. Seo, J. Seong, and J. Bae, “Mitigation of crosstalk errors in a quantum measurement and its applications,” 2021.

[43] T. Alexander *et al.*, “Qiskit pulse: programming quantum computers through the cloud with pulses,” *Quantum Science and Technology*, vol. 5, no. 4, p. 044006, 2020.

[44] Rigetti, “Quil-t: an extension for the pulse-level control of quantum programs,” https://pyquil-docs.rigetti.com/en/stable/quilt.html [accessed: 2022.01.24].

[45] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Physical review letters*, vol. 119, no. 18, p. 180509, 2017.

[46] R. Harper and S. T. Flammia, “Fault-tolerant logical gates in the ibm quantum experience,” *Phys. Rev. Lett.*, vol. 122, p. 080504, 2019.

[47] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles, “Machine learning of noise-resilient quantum circuits,” *PRX Quantum*, vol. 2, no. 1, p. 010324, 2021.

[48] A. R. Carvalho, H. Ball, M. J. Biercuk, M. R. Hush, and F. Thomsen, “Error-robust quantum logic optimization using a cloud quantum computer interface,” *Physical Review Applied*, vol. 15, no. 6, p. 064054, 2021.

[49] T. Giurgica-Tiron, Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, “Digital zero noise extrapolation for quantum error mitigation,” in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2020, pp. 306–316.

[50] D. Bacon, “Operator quantum error-correcting subsystems for self-correcting quantum memories,” *Phys. Rev. A*, vol. 73, p. 012340, Jan 2006.

[51] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Phys. Rev. A*, vol. 54, pp. 1098–1105, Aug 1996.

[52] A. Steane, “Multiple-particle interference and quantum error correction,” *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 452, no. 1954, pp. 2551–2577, 1996.

[53] F. B. Maciejewski, F. Baccari, Z. Zimborás, and M. Oszmaniec, “Modeling and mitigation of realistic readout noise with applications to the quantum approximate optimization algorithm,” *arXiv preprint arXiv:2101.02331*, 2021.

[54] C. Bauer, A. He, W. A. de Jong, B. Nachman, and V. R. Pascuzzi, “Computationally efficient zero noise extrapolation for quantum gate error mitigation,” *arXiv preprint arXiv:2110.13338*, 2021.

[55] IBM, “Overview of qiskit runtime,” https://github.com/Qiskit-Partners/qiskit-runtime [accessed: 2022.01.24].

[56] A. Gilliam *et al.*, “Foundational patterns for efficient quantum computing,” *arXiv preprint arXiv:1907.11513*, 2019.

[57] Y. Huang and M. Martonosi, “Statistical assertions for validating patterns and finding bugs in quantum programs,” in *Proceedings of the 46th International Symposium on Computer Architecture*, 2019, pp. 541–553.

[58] S. Perdrix, “Quantum patterns and types for entanglement and separability,” *Electronic Notes in Theoretical Computer Science*, vol. 170, pp. 125–138, 2007.

[59] G. Hohpe and B. Woolf, “Enterprise integration patterns,” in *9th conference on pattern language of programs*, 2002, pp. 1–9.

[60] M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, and F. Leymann, “Efficient pattern application: Validating the concept of solution implementations in different domains,” vol. 7, no. 3&4. Xpert Publishing Services (XPS), 2014, pp. 710–726.

[61] M. Falkenthal *et al.*, “Leveraging pattern application via pattern refinement,” in *Proceedings of the International Conference on Pursuit of Pattern Languages for Societal Change (PURPLSOC)*, 2016, pp. 38–61.

[62] M. M. Hafidhi and E. Boutillon, “Hardware error correction using local syndromes,” in *2017 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.

[63] L.-J. Saiz-Adalid *et al.*, “Reducing the overhead of bch codes: New double error correction codes,” *Electronics*, vol. 9, no. 11, p. 1897, 2020.

[64] D. MacKay, G. Mitchison, and P. McFadden, “Sparse-graph codes for quantum error correction,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2315–2330, 2004.

[65] PlanQK, “Planqk - the platform for quantum-supported artificial intelligence,” https://platform.planqk.de/ [accessed: 2022.01.24].

[66] F. Leymann and J. Barzen, *Pattern Atlas*. Springer International Publishing, 2021, pp. 67–76.

[67] PlanQK, “Planqk - integration of the patternrepository pattern atlas,” https://patterns.platform.planqk.de/pattern-languages [accessed: 2022.01.24].

Contributions by Feature Layers in Two-Class Deep Learning Image Classification Decisions

Debanjali Banerjee
School of Computing and Informatics
University of Louisiana at Lafayette
U.S.A.
Email: debanjali.banerjee1@louisiana.edu

Chee-Hung Henry Chu
School of Computing and Informatics
University of Louisiana at Lafayette
U.S.A.
Email: chu@louisiana.edu

***Abstract*—Deep learning methods have excellent accuracy achievements in image classification but largely remains a black box method. Image classification is the core of many machine vision tasks, including object detection. Better understanding of how the classification decision is made will improve the understanding of such tasks as object detection. In this work, we train a deep learning network to classify between two classes. We compute the so-called SHapley Additive exPlanations (SHAP) values for the feature layers using input images against a population of other training images for the classification layer. The SHAP value is a special case of the Shapley value which explains the factors in a machine learning decision by measuring the output change due to change in each factor. The SHAP value is the Shapley value satisfying local accuracy, missingness, and consistency properties. Experimental results show the different responses from the lowest to the highest feature extraction layers.**

***Index Terms*—Deep Learning, Explainable Artificial Intelligence, Shapley Values, Image Classification.**

I. Introduction

Image classification was the task that sparked the research interest in deep learning for the past ten years. In image classification, an image is classified to one of several classes. Instead of using custom crafted feature vectors, a deep learning approach uses different, increasingly complex convolutional networks to extract images features as input to a classification layer. Image classification forms the core of more complex artificial intelligence systems such as object detection. As such, a better understanding of the image classification task might lead to better understanding of such tasks as object detection. Within three years of the publication of the AlexNet [1], deep learning systems were already shown to perform better than humans in image classification. Despite such advances, deep learning systems remain black boxes and little is known as to how they make decisions.

A number of recent works have been reported in developing methods that address how to explain complex machine learning systems. A recent survey [2] highlighted domain-dependent and context-specific methods for dealing with the interpretation of artificial intelligence systems, including the boundaries and gaps of recent advances. One approach is to remove a feature and then assessing the output change. The complications in computing are due to the large number of permutations involved in the remaining features that must be averaged to determine the Shapley values. In [3] the Shapley values were optimized to include the local accuracy, missingness, and consistency properties to form the so-called SHapley Additive exPlanations (SHAP) values. A different approach [4] is the synthesis of Local Interpretable Model-agnostic Explanations (LIME). Other methods that focus on the image space [5] often lead to solutions that overlap with image saliency research [6].

In our work, we are interested in determining the contributions of the features in the large hidden layers. We compute the SHAP values to understand the contributions of individual features not at the image level but at the hidden, feature levels. One reason for doing so is to validate that all features in a deep learning network make contributions to decisions.

In Section II, we describe the use of Shapley values to explain the contributions of individual features in a deep learning network. In Section III, we describe our experiments and the results. Finally, we draw our conclusion in Section IV.

II. Methodology

We assess the contributions of feature values in a deep learning network that has been trained to perform image classification. We describe our approach in assessing the contributions of the feature extraction stage of a deep learning network in Section II-A. We compute the SHAP values, which is a special case of the Shapley values. In Section II-B we present an overview of the Shapley values that we use for calculating the contributions.

A. Deep Learning Features

A convolutional neural network broadly speaking has an input layer, convolutional layers, max or average pooling layers, fully connected layers, and an output layer. The convolutional and max pooling layers are typically in the feature extraction stage while the fully connected layers and an output layer are in the classification stage. A convolutional layer takes as input a block of feature values $F^l(i,j,k)$ where l is the layer index, $i = 0,\cdots,M_l - 1$, $j = 0,\cdots,N_l - 1$, and $k = 0,\cdots,K_l - 1$, so that the block of feature values is of size $M_l \times N_l \times K_l$; typically $M_l = N_l$. The input feature block is transformed by a bank of convolutional kernels and associated nonlinearity to produce an output block $F^{l+1}(i,j,k)$ of size $M_{l+1} \times N_{l+1} \times K_{l+1}$ where M_{l+1}, N_{l+1} are controlled by the strides of the convolution step and K_{l+1} is controlled by the number of convolutional kernels used.

In machine learning, we use a training data set to determine the convolutional kernel coefficient values in a deep learning network. After training, when we apply an image as input, the feature values are calculated at all levels and an output decision is made. A question is: what is the contribution of a particular feature value in the decision? Our approach to answering this is to compute the SHAP value for the feature values $F^l(i,j,k)$ $i = 0,\cdots,M_l - 1$, $j = 0,\cdots,N_l - 1$, $k = 0,\cdots,K_l - 1$, and l for the layers in the feature extraction stage.

B. *Shapley and SHAP Values*

Consider a simple example where there are four inputs x_0, x_1, x_2, and x_3 to a system f that provides an output $y = f(x_0,x_1,x_2,x_3)$. Suppose we would like to assess the contribution of x_0. An intuitive way to do so is to turn off x_0 and consider the change—either an increase or a decrease—to the output value. Let $y_0 = f(0,x_1,x_2,x_3)$ and $\delta_0 = y - y_0$. We can repeat this process to find δ_i for $i = 1,2,3$. The value δ_i would be an indicator of the contribution of x_i to the output y. When a training set is available, we feed the entire data set and average the output, so that $\overline{\delta_0} = \overline{y_0} - \bar{y}$, where $\overline{y_0}$ and $\bar{y}$ are the outputs with x_0 turned off and on, respectively, averaged over the data set. This intuitive example, however, does not consider the other cases, such as when other combinations of features are turned off.

A more comprehensive approach as illustrated by our example is as follows. Let the set of all features be F; in our example $F = \{x_0,x_1,x_2,x_3\}$. If we exclude an input, say x_0, from F, to study the contribution of x_0 to the output, then we need to consider all combinations of the remaining features in $F\backslash\{x_0\}$. The set of all such combinations is the power set of $F\backslash\{x_0\}$, $2^{F\backslash\{x_0\}} = \{\phi,\{x_1\},\{x_2\},\{x_3\},\{x_1,x_2\},\{x_1,x_3\},\{x_2,x_3\},\{x_1,x_2,x_3\}\}$. Let $S \in 2^{F\backslash\{x_0\}}$ so that S is a subset of features excluding x_0. Let f_S be the output f trained using the subset of features S. The difference $f_{S\cup\{x_0\}} - f_S$ is a measure of how much the output changes with the inclusion of x_0 relative to S, a subset of the features. We can then compute a weighted sum of the change in output over all possible subsets of $F\backslash\{x_0\}$. The weights are set such that they sum to 1 and that all weights associated with subsets with the same cardinality are equal. Let $C(n,k)$ denote the number of combinations in choosing k from n items. There are $C(|F|,|S\cup\{x_0\}|) = C(|F|,|S|+1)$ combinations of choosing subsets $S\cup\{x_0\}$ from F. Each such combination has $|S|+1$ terms so that the total number of terms in the sum is $(|S|+1)C(|F|,|S|+1)$. The change in output is then weighted by the reciprocal of the number of terms

$$\frac{1}{(|S|+1)C(|F|,|S|+1)}.$$

The weight expands to

$$\frac{(|S|+1)!(|F|-|S|-1)!}{(|S|+1)|F|!} = \frac{|S|!(|F|-|S|-1)!}{|F|!},$$

so that the weighted average is given by

$$\sum_{S\subset F\backslash\{x_0\}} \frac{|S|!(|F|-|S|-1)!}{|F|!}(f_{S\cup\{x_0\}} - f_S).$$

We note that to calculate the Shapley value for one of the $|F|$ features, we need to build $2^{|F|}$ different models.

The SHAP value is a special case of the Shapley value satisfying local accuracy, missingness, and consistency properties. It uses a number of so-called "Explainers" to possibly approximate the values for determining the contributions by feature values.

III. Experimental Results

We ran experiments to validate our methodology of determining the contributions of feature layers in a two-class deep learning image classifier. We describe our data set in Section III-A. We picked a standard image classifier, viz. the VGG 19 network, as described in Section III-B. We describe how we compute the SHAP values in Section III-C and show the results in Section III-D.

A. *Data Set*

We used the data set from the Kaggle cat-vs-dog competition [7]. The goal is to distinguish between a cat vs. a dog in the given input image. The images were of different aspect ratios, different sizes, and most images include background. Examples are shown in Figure 1. We used a balanced training set of 2,000 cats and 2,000 dogs to train the classifier layer. We used data augmentation that includes rotation and scaling.

B. *The Classifier*

We trained a VGG 19-based network [8] with the top classification layer replaced by a two-class classifier (Table I). Though the VGG network architecture is well-known, we include the summary here to refer to the layer levels in Section III. The classification layer takes the 512×7×7 tensor of the features and feeds them to a layer of 64 hidden units and then to two output units, corresponding to the two classes. We used the ReLU nonlinearity in all neurons, except for the output units. We used transfer learning so that the lower, feature extraction layers with 20,024,384 weights were trained with the ImageNet data set. The top classification layers with 1,605,826 weights were trained with the application-specific data set of cats and dogs. We used dropout in training. The training accuracy was around 95% while the state-of-the-art was above 98%. Our goal here was not in achieving the ultimate in accuracy but to obtain a "good enough" architecture for analysis of the contributions of each of the feature layers.

TABLE I: SUMMARY OF THE VGG 19 NETWORK FOR CLASSIFICATION.

Layer Number	Type	Output Shape
1	InputLayer	[(None, 224, 224, 3)]
2	Conv2D	(None, 224, 224, 64)
3	Conv2D	(None, 224, 224, 64)
4	MaxPooling2D	(None, 112, 112, 64)
5	Conv2D	(None, 112, 112, 128)
6	Conv2D	(None, 112, 112, 128)
7	MaxPooling2D	(None, 56, 56, 128)
8	Conv2D	(None, 56, 56, 256)
9	Conv2D	(None, 56, 56, 256)
10	Conv2D	(None, 56, 56, 256)
11	Conv2D	(None, 56, 56, 256)
12	MaxPooling2D	(None, 28, 28, 256)
13	Conv2D	(None, 28, 28, 512)
14	Conv2D	(None, 28, 28, 512)
15	Conv2D	(None, 28, 28, 512)
16	Conv2D	(None, 28, 28, 512)
17	MaxPooling2D	(None, 14, 14, 512)
18	Conv2D	(None, 14, 14, 512)
19	Conv2D	(None, 14, 14, 512)
20	Conv2D	(None, 14, 14, 512)
21	Conv2D	(None, 14, 14, 512)
22	MaxPooling2D	(None, 7, 7, 512)
23	Flatten	(None, 25088)
24	Dense	(None, 64)
25	Dropout	(None, 64)
26	Dense	(None, 2)

C. SHAP Calculation

We used the publicly available SHAP package [9] in the Python environment. The SHAP package needs a sub-population to average over. We randomly selected 25 cat images and 25 dog images from the training set. We iteratively computed the SHAP values of the feature layer tensors from Layer 1 to Layer 22, which made up the feature extraction layers of the trained network. Each feature layer was a tensor with different spatial resolutions and depths that depended on the block. We visualized the SHAP values by scaling both the values and the spatial resolution to the input image. It was not practical to display the many different individual feature maps in a tensor; we summed the values across channels in a tensor to form a composite feature map.

D. Results

We used four input images for classification as shown in Figure 1. All four images were correctly classified by the deep learning classifier. Each image was fed as input to the SHAP values calculation individually; i.e., they were not processed as a batch. Hence when, e.g., "cat3238" was used as the input, the other 3 images were not processed by the network. We loaded the network from its trained state for every round of SHAP values calculation.

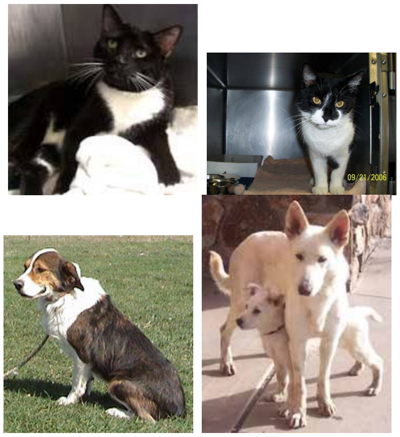
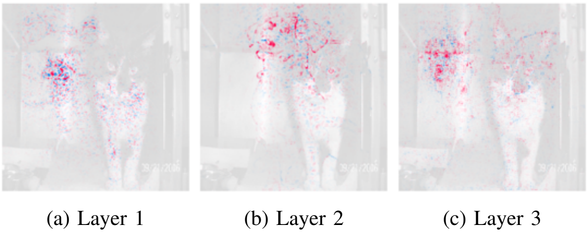


Fig. 1: Input pictures for classification "cat3238" (top left); "cat3333" (top right); "dog3333" (lower left); "dog3399" (lower right).

We show the Shapley values for the feature map tensors at each layer when the input was "cat3333". In the following, the negative values are mapped to blue while the positive values are mapped to red. We overlay the SHAP values on a grayscale version of the input image.

In Figure 2, we show the feature maps when the spatial resolution were 224×224. In layers 2 and 3 we can see outlines of dogs in parts of the image that had very little color variation. Next we show the feature maps when the spatial resolution were 112×112 in Figure 3. The outlines of dogs can be seen in layers 4 and 5 as well.



(a) Layer 1 (b) Layer 2 (c) Layer 3

Fig. 2: SHAP values for layers 1, 2, and 3 when the input was "cat3333".

The feature maps when the spatial resolution were 56×56 in Figure 4. At this resolution, the high values

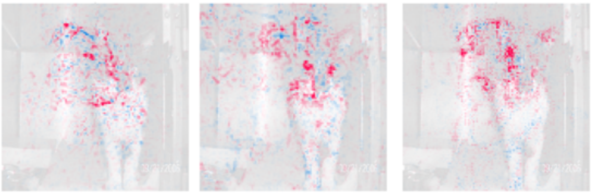
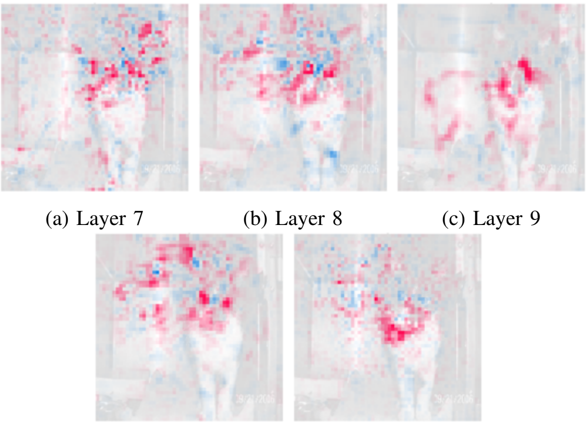

(a) Layer 4 (b) Layer 5 (c) Layer 6

Fig. 3: SHAP values for layers 4, 5, and 6 when the input was "cat3333".

were better localized to the cat in the input image, even though there were still responses to the area (left of the cat in the image) with low color variations. The feature maps when the spatial resolution were 28×28 and 14×14 in figures 5 and 6, respectively. We show the feature map at the top of the feature extraction layers when the spatial resolution was 7×7, in Figure 7. We can see that at the lower spatial resolution, the "shape" information is less evident and the focus was more on the face of the cat in the input image.

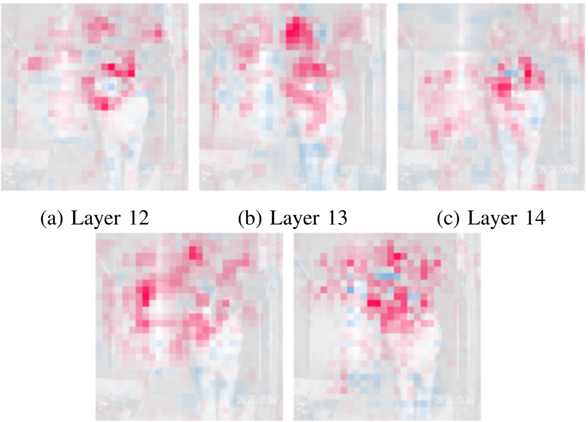

(a) Layer 7 (b) Layer 8 (c) Layer 9

(d) Layer 10 (e) Layer 11

Fig. 4: SHAP values for layers 7 to 11 when the input was "cat3333".

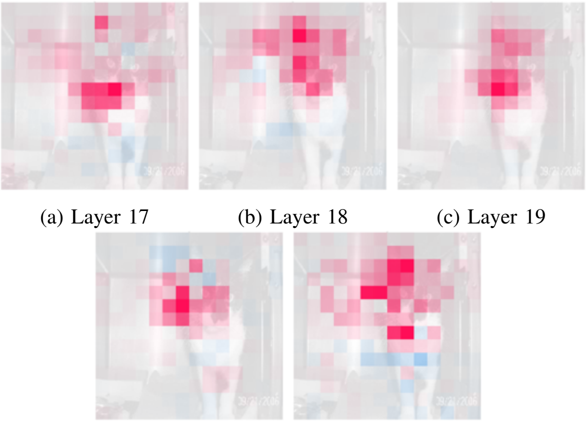
We repeated the calculations for the other 3 input images but do not show the full sets of results here in the interest of brevity. We wanted to explore the phenomenon of having "imagined" faces in the lower layers. In figures 8, 9, and 10, we show the responses at layers 2, 3, 5, and 6 for the other three input images. In all figures, we saw some outlines of other faces, some stronger (Figure 8) and some less (Figure 10).

Given that we used transfer learning, we next compared the input images to images that were in the ImageNet data set that was used to train the feature extraction layers. We replaced the 50 cat and dog images by 50 images randomly selected from the ImageNet database in the SHAP value computations. In Figure 11, we show the SHAP values at layers 2, 3, 5, and 6. While we saw some responses again in the background area with little color variation, we did not see the same face outline that we saw in Figure 2. or 3. In Figure 12, we show the SHAP values at the layers right before the spatial resolution was reduced in half.


(a) Layer 12 (b) Layer 13 (c) Layer 14

(d) Layer 15 (e) Layer 16

Fig. 5: SHAP values for layers 12 to 16 when the input was "cat3333".


(a) Layer 17 (b) Layer 18 (c) Layer 19

(d) Layer 20 (e) Layer 21

Fig. 6: SHAP values for layers 17 to 21 when the input was "cat3333".

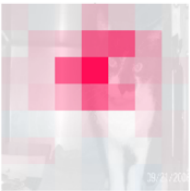
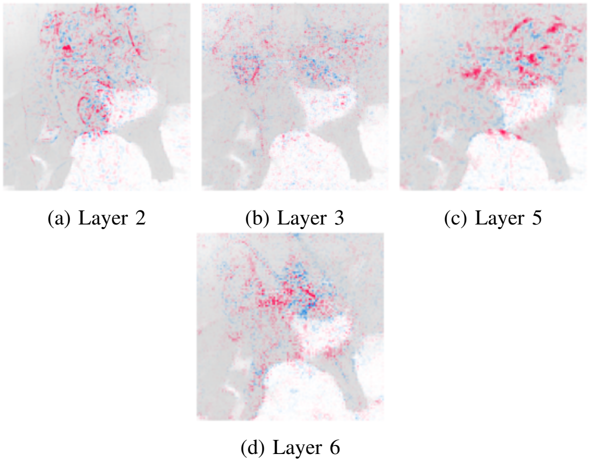


Fig. 7: SHAP values for Layer 22 when the input was "cat3333".

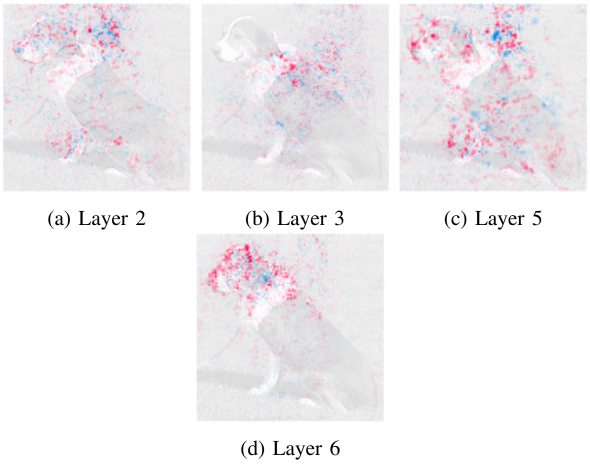
We can see that by comparing the input image to the ImageNet images in computing the SHAP values, the



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

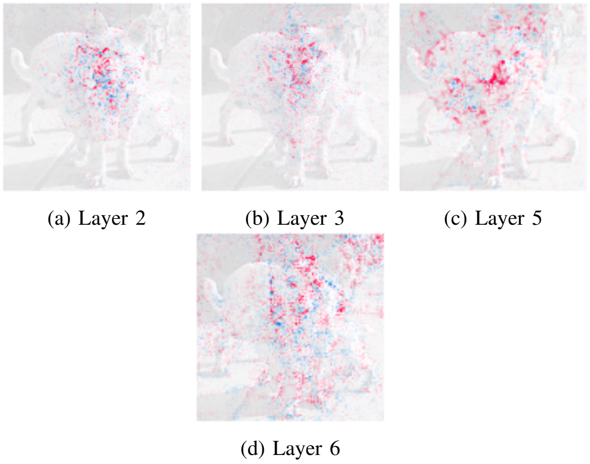
Fig. 8: SHAP values for layers layers 2, 3, 5, and 6 when the input was “cat3238”.



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 9: SHAP values for layers layers 2, 3, 5, and 6 when the input was “dog3333”.

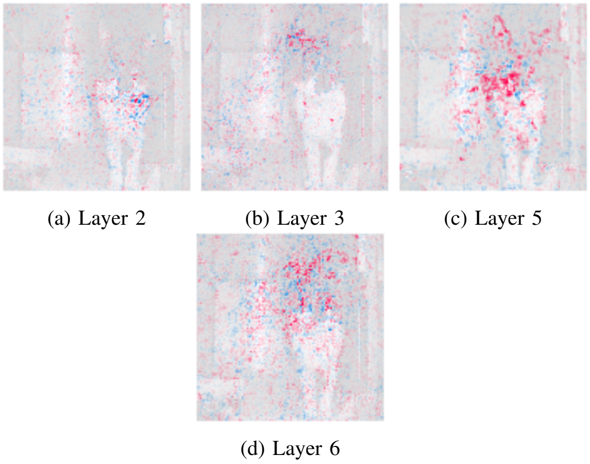


(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 10: SHAP values for layers 2, 3, 5, and 6 when the input was “dog3399”.

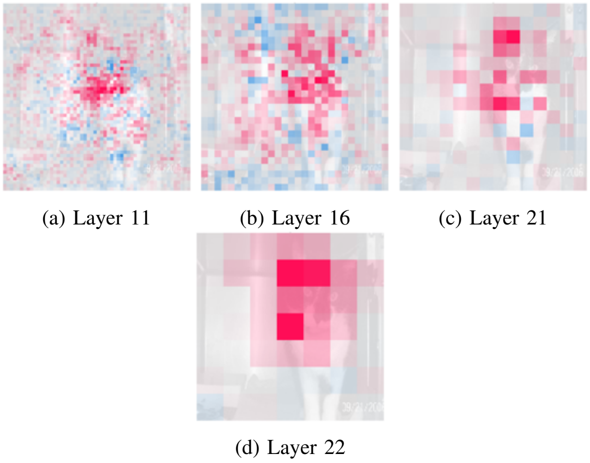
high values correspond more to the outline of the input image. The evidence is that the outline artifacts seen, e.g., in Figure 2 or 3 were influenced by the classification layers.



(a) Layer 2 (b) Layer 3 (c) Layer 5

(d) Layer 6

Fig. 11: SHAP values calculated using ImageNet images for comparison for layers 2, 3, 5, and 6 when the input was “cat3333”.



(a) Layer 11 (b) Layer 16 (c) Layer 21

(d) Layer 22

Fig. 12: SHAP values calculated using ImageNet images for comparison for layers 11, 16, 21 , and 22 when the input was “cat3333”.

IV. Conclusion and Future Work

We computed the SHAP values of the layered feature tensors in a deep learning network trained to distinguish between two classes. The SHAP values are a special case of the Shapley value that explains the factors in a machine learning decision by measuring the output change due to change in each factor. The SHAP value is the Shapley value satisfying local accuracy, missingness, and consistency properties. Our results showed that the lower layers exhibited shapes that fit other faces, some of them not even from the same class. It appeared that the network worked on assembling the outlines of a shape much earlier in the layered architecture than expected, as early as Layer 2 which was immediately connected to the input layer.

There are a number of interesting directions of future work. In the present work, we examined cases in which the network made the correct classification decision. Given that the network accuracy is 95%, there are cases when the network misclassifies. Ongoing work using the same network architecture and the same data set includes running similar analyses for misclassified samples. We would also like to calculate the SHAP values for the top classification layer. When the network makes a decision, we would like to investigate creating a tree of high SHAP values to attempt to explain the decision. Along other directions, we would like to generalize our findings to different data sets and network architectures.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers whose suggestions helped to improve the clarity of our paper. This work was supported in part by the U.S. National Science Foundation under grant number OIA-1946231 and the Louisiana Board of Regents for the Louisiana Materials Design Alliance (LAMDA).

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2012.

[2] G. Vilone and L. Longo, “Notions of explainability and evaluation approaches for explainable artificial intelligence,” *Information Fusion*, vol. 76, pp. 89–106, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001093

[3] S. Lundberg and S.-I. Lee. (2017) A unified approach to interpreting model predictions. Last accessed: 15 April 2022. [Online]. Available: https://arxiv.org/abs/1705.07874

[4] F. Stieler, F. Rabe, and B. Bauer, “Towards domain-specific explainable ai: Model interpretation of a skin image classifier using a human approach,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 1802–1809.

[5] M. Ghassemi, L. Oakden-Rayner, and A. L. Beam, “The false hope of current approaches to explainable artificial intelligence in health care,” *The Lancet Digital Health*, vol. 3, pp. e745–e750, 2021.

[6] A. Singh, C. H. Chu, and M. A. Pratt, “Comparing color descriptors between image segments for saliency detection,” in *Proceedings of the International Conference on Pattern Recognition Applications and Methods*, Rome, Italy, 2 2016, pp. 558–565.

[7] Kaggle. (2013) Dogs vs. cats: Create an algorithm to distinguish dogs from cats. Last accessed: 15 April 2022. [Online]. Available: https://www.kaggle.com/c/dogs-vs-cats

[8] K. Simonyan and A. Zisserman. (2014) Very deep convolutional networks for large-scale image recognition. Last accessed: 15 April 2022. [Online]. Available: https://arxiv.org/abs/1409.1556

[9] S. Lundberg. (2020) A game theoretic approach to explain the output of any machine learning model. Last accessed: 15 April 2022. [Online]. Available: https://github.com/slundberg/shap